



FACULTAD DE INGENIERÍA
ESCUELA DE SISTEMAS

**DISERTACIÓN DE GRADO PREVIA LA OBTENCIÓN DEL TÍTULO DE
INGENIERO EN SISTEMAS Y COMPUTACIÓN**

TEMA:

**DESARROLLO DE UN SISTEMA WEB PARA ADMINISTRACIÓN DE
EVENTOS Y CONTROL DE ENTRADA Y SALIDA DE EMPLEADOS
CASO DE ESTUDIO: CENTRO DE FORMACIÓN SOCIAL "BETHANIA"**

DIRECTOR:

ING. BEATRIZ CAMPOS

AUTOR:

GUSTAVO WLADIMIR MOYANO MEJÍA

QUITO – 2015

Declaratoria de Responsabilidad

Los conceptos desarrollados, análisis realizados y las conclusiones del presente trabajo, son de exclusiva responsabilidad del autor

Gustavo Wladimir Moyano Mejía

Dedicatoria y Agradecimiento

Dedicado a esa fuerza que es guía y es camino, que mueve mis pies, manos y mente.

Dedicatoria y agradecimientos sinceros y totales a Gustavo Xavier Moyano Álvarez y Sandra Edith Mejía Peñafiel ¡por ser inspiración, impulso, ejemplo, apoyo diario y mano dura!

A Yolanda Alicia Álvarez, María Angélica Peñafiel y Patricio Hernán Izurieta, por ser enseñanza, sabiduría y cariño.

Y a Stephanie Mishell Suasnavas Garrido, por ser amor, incondicionalidad y esperanza.

“Success is not final, failure is not fatal: It is the courage to continue that counts.”

-Winston Churchill

Resumen

El presente documento expone el enfoque tomado para el desarrollo de un sitio web de administración de eventos, desde la metodología tomada para el desarrollo, las herramientas utilizadas, hasta la puesta en marcha del sistema en el centro de formación “*Bethania*”.

La metodología de software utilizada, “*Disciplined Agile Delivery*”, ofrece un marco de desarrollo de software ágil, basándose en diferentes técnicas de modelamiento, de ciclo de vida y de desarrollo de software (*Extreme Programming, Scrum Kanban, etc.*), para llevar a cabo proyectos en grandes empresas. La metodología utiliza la filosofía del “agilismo” para definir sus principios, una tendencia aplicada cada vez con más frecuencia entre desarrolladores de software.

El sistema desarrollado utiliza una arquitectura Web basada en Modelo-Vista-Controlador, y se vale de herramientas como Java Persistence API, Java RESTful services, Angular JS, HTML5 y CSS3, que permiten tener gran dinámica en el viaje de datos y optimiza el rendimiento de la aplicación balanceando cargas de trabajo entre cliente y servidor.

El sistema facilita el proceso actual de administración de eventos manejada por el personal administrativo de Bethania, lo que permite manejar el tiempo eficientemente y abre puertas para utilizar el tiempo ahorrado en diferentes tareas.

Durante todo el desarrollo del proyecto, la cercanía con el cliente es lo que ha permitido generar una solución consumible que cubra la mayor cantidad de necesidades de una manera óptima, por lo que ésta característica es una de las más resaltables de la filosofía de desarrollo ágil.

En conclusión, la aplicación de las metodologías ágiles fusionadas con herramientas de desarrollo de software de vanguardia, conjuntamente con la cercanía frecuente con el cliente, permiten generar una solución automatizada óptima, desde su proceso de creación hasta el tiempo que permite ahorrar ocupándose de tareas triviales iterativas.

Índice de Contenidos

Declaratoria de Responsabilidad.....	2
Dedicatoria y Agradecimiento.....	3
Resumen.....	4
Índice de Figuras	8
1. CAPÍTULO 1 – INTRODUCCIÓN Y FUNDAMENTO TEÓRICO	10
1.1. Objetivos	10
1.1.1. Objetivo General.....	10
1.1.2. Objetivos Específicos	10
1.2. Metodología Disciplined Agile Delivery	11
1.2.1. Introducción.....	11
1.2.2. Ciclos de Vida de Entrega Completamente Ágil.....	14
1.2.3. Enfoque guiado por Metas - Metas de Proceso	23
1.2.4. Conciencia Empresarial.....	27
1.2.5. Soluciones “Consumibles”	30
1.2.6. El Manifiesto Ágil Disciplinado (“The Disciplined Agile Manifesto”)	31
1.2.7. Ventajas y Desventajas de la Metodología	33
1.3. Herramientas de Desarrollo	34
1.3.1. Java Persistence Application Programming Interface (JPA)	34
1.3.2. Hibernate.....	34
1.3.3. Java RESTful Services	35
1.3.4. Javascript	36
1.3.5. Angular JS.....	37
1.3.6. HTML5.....	39
1.3.7. CSS 3.....	40
2. CAPÍTULO 2 - CASO DE ESTUDIO	41
2.1. Giro del Negocio.....	41
2.1.1. Datos de la Organización	41
2.1.2. Misión	42
2.1.3. Visión	42
2.1.4. Objetivos	42

2.2.	Situación Actual.....	42
2.3.	Levantamiento de Procesos	44
2.3.1.	Proceso Clave	45
2.3.2.	Procesos de Apoyo.....	48
2.3.3.	Procesos de Dirección.....	48
3.	CAPÍTULO 3 - SOLUCIÓN PROPUESTA / APLICACIÓN DAD.....	49
3.1.	Ajuste de la Metodología	49
3.1.1.	Definición de Metas	49
3.1.2.	Definición de ciclo de vida	50
3.2.	Iteración 0 – Concepción	51
3.2.1.	Desarrollo de una visión común.....	53
3.2.2.	Explorar un alcance inicial	54
3.2.3.	Identificar estrategia técnica inicial	72
3.2.4.	Desarrollar un plan de liberación inicial.....	75
4.	CAPÍTULO 4 – CONCLUSIONES Y RECOMENDACIONES.....	77
4.1.	Conclusiones.....	77
4.2.	Recomendaciones.....	79
	BIBLIOGRAFÍA.....	80
	ANEXOS.....	81

Índice de Figuras

FIGURA 1 UN MARCO DE TRABAJO HÍBRIDO	12
FIGURA 2 CICLO DE VIDA DE ENTREGA COMPLETA.....	13
FIGURA 3 IMPULSADO POR METAS	13
FIGURA 4 CICLO DE VIDA DE ENTREGA COMPLETAMENTE ÁGIL.....	15
FIGURA 5 CICLO DE VIDA DE ENTREGA COMPLETAMENTE ÁGIL – SIGUIENTE NIVEL	16
FIGURA 6 CICLO DE VIDA DAD EXTENDIENDO A SCRUM.....	18
FIGURA 7 CICLO DE VIDA DAD AVANZADO / AJUSTADO.....	20
FIGURA 8 CICLO DE VIDA DAD ENTREGA CONTINUA	21
FIGURA 9 CICLO DE VIDA DAD EXPLORATORIO	22
FIGURA 10 METAS DE PROCESO	24
FIGURA 11 EJEMPLO DE DIAGRAMA DE META DE PROCESO.....	25
FIGURA 12 CINCO NIVELES DE CONCIENCIA.....	28
FIGURA 13 DIAGRAMA DE PROCESOS	44
FIGURA 14 PROCESO DE RESERVACIÓN DE EVENTOS	45
FIGURA 15 DIAGRAMA DE FLUJO RESERVACIÓN DE EVENTOS.....	46
FIGURA 16 PROCESO DE PREPARACIÓN DE EVENTO.....	47
FIGURA 17 PROCESO DE PUESTA EN MARCHA DE EVENTO	47
FIGURA 18 METAS DE CONCEPCIÓN DAD.....	49
FIGURA 19 METAS DE CONSTRUCCIÓN DAD.....	50
FIGURA 20 METAS DE TRANSICIÓN DAD.....	50
FIGURA 21 META DESARROLLO DE UNA VISIÓN COMÚN.....	53
FIGURA 22 META EXPLORAR UN ALCANCE INICIAL	54
FIGURA 23 FRENTES DE TRABAJO	55
FIGURA 24 CASO DE USO GENERAL.....	56
FIGURA 25 CASO DE USO 1 - ADMINISTRACIÓN EVENTOS	57
FIGURA 26 CU1 - SIGUIENTE NIVEL 1 - RESERVAR EVENTO.....	57
FIGURA 27 CU1 - SIGUIENTE NIVEL 2 - CREAR EVENTO.....	59
FIGURA 28 CU1 - SIGUIENTE NIVEL 3 - MODIFICAR EVENTO	60
FIGURA 29 CU1 - SIGUIENTE NIVEL 4 - ELIMINAR EVENTO	61
FIGURA 30 CU1 - SIGUIENTE NIVEL 5 - CONSULTA DATOS GENERALES EVENTO	62
FIGURA 31 CU1 - SIGUIENTE NIVEL 6 - CONSULTA FILTRO TIEMPO EVENTO....	62
FIGURA 32 CASO DE USO 2 - ADMINISTRACIÓN DE CLIENTES.....	63
FIGURA 33 DEFINICIÓN DE PERSONAS – ADMINISTRADORA	64
FIGURA 34 DEFINICIÓN DE PERSONAS – COLABORADOR.....	64
FIGURA 35 HISTORIA DE USUARIO 1 - ADMINISTRACIÓN TRABAJADORES	65
FIGURA 36 HISTORIA DE USUARIO 2 - CONTROL TIEMPOS E/S.....	65
FIGURA 37 ADMIN EVENTOS - RELACIÓN DE ENTIDADES.....	67
FIGURA 38 ADMIN EVENTOS - DIAGRAMA CONCEPTUAL DE BASE DE DATOS..	68

FIGURA 39 ADMIN EVENTOS - DIAGRAMA FÍSICO DE BASE DE DATOS.....	68
FIGURA 40 CONTROL E/S - RELACIÓN DE ENTIDADES.....	69
FIGURA 41 CONTROL E/S DIAGRAMA CONCEPTUAL DE BASE DE DATOS.....	69
FIGURA 42 CONTROL E/S DIAGRAMA FÍSICO DE BASE DE DATOS	69
FIGURA 43 TIPOS DE ITEMS DE TRABAJO.....	70
FIGURA 44 FUNCIONAMIENTO DE LISTA DE ITEMS DE TRABAJO.....	71
FIGURA 45 META IDENTIFICAR ESTRATEGIA TÉCNICA INICIAL	72
FIGURA 46 ARQUITECTURA DE APLICACIÓN	73
FIGURA 47 MAPA WEB DE APLICACIÓN.....	74
FIGURA 48 PROTOTIPO DE CALENDARIO DE EVENTOS.....	74
FIGURA 49 META DESARROLLAR UN PLAN DE LIBERACIÓN INICIAL	75

1. CAPÍTULO 1 – INTRODUCCIÓN Y FUNDAMENTO TEÓRICO

1.1. Objetivos

1.1.1. Objetivo General

- Desarrollar un sistema para la administración de eventos, control de asistencia de trabajadores y promoción de servicios que ofrece el centro de formación social BETHANIA.

1.1.2. Objetivos Específicos

- Levantar y analizar requerimientos necesarios para el desarrollo del sistema.
- Analizar y describir la metodología del diseño a utilizar.
- Realizar el diseño lógico y físico de los procesos a automatizar.
- Desarrollar una solución web para la administración de eventos y clientes, utilizando software libre, con una interface amigable para usuarios.
- Agregar módulo de reportes estadísticos anuales de eventos con visualización gráfica.
- Desarrollar una aplicación de escritorio para el control de asistencia de trabajadores utilizando software libre.
- Implementar la solución web y la aplicación, realizando pruebas respectivas.
- Documentar la información técnica, de usuario final y de capacitación.

1.2. Metodología Disciplined Agile Delivery

1.2.1. Introducción

La metodología Disciplined Agile Delivery (DAD) puede encajar dentro de la siguiente definición: “El marco de trabajo y de toma de decisiones DAD es un enfoque ágil, híbrido, orientado al aprendizaje y que toma en cuenta primero al talento humano, de entrega de soluciones de tecnologías de la información.”¹

Principalmente la metodología DAD extiende a Scrum² añadiéndole estrategias como: Agile Modeling (AM), Extreme Programming (XP), Unified Process, Kanban, Lean Software Development y Outside In Development (OID); permitiendo a DAD titularse una metodología híbrida para la entrega de soluciones de TI. DAD extiende también el ciclo de vida de Scrum (enfocado en la construcción) para hacer frente completamente al proyecto desde su etapa inicial hasta la entrega del producto. A diferencia de otras metodologías ágiles, DAD no se atiene a un solo ciclo de vida en especial, sino es flexible y apto para el cambio continuo.

DAD incluye las de prácticas técnicas de Extreme Programming (XP), y las estrategias de modelado, documentación y gobernación que no están presentes en Scrum ni en XP, construyendo un enfoque sólido pero flexible.

Un proyecto implementado bajo DAD es guiado por Metas u Objetivos (Hitos) a cumplir, en donde se pueden visualizar alternativas de solución para un problema. Esto permitirá entallar a la metodología para que se ajuste a las necesidades específicas del proyecto y permita decidir cómo llegar a una solución efectiva y funcional para el cliente.

¹ Definición tomada de <https://disciplinedagiledelivery.wordpress.com>

² Marco de trabajo para la gestión y desarrollo de software basada en un proceso iterativo e incremental utilizado comúnmente en entornos basados en el desarrollo ágil de software

1.2.1.1. *Un Marco de Trabajo Híbrido*

Disciplined Agile Delivery (DAD) es un marco de trabajo construido sobre una base sólida constituida por varias disciplinas combinadas a manera de piezas, siendo DAD el rompecabezas completo.

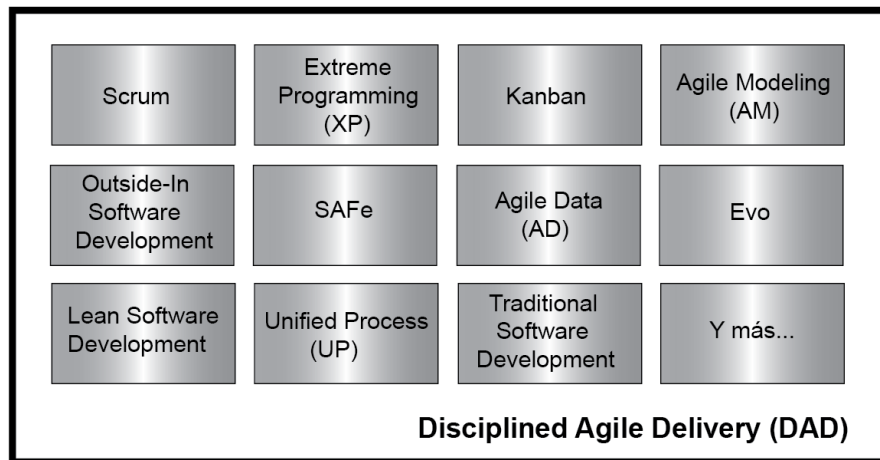


FIGURA 1 Un Marco de Trabajo Híbrido

Autores: Scott W Ambler - Disciplined Agile Consortium

Ésta característica es una ventaja al momento de manejar DAD, su riqueza de técnicas, prácticas y estrategias disponibles para ser utilizadas.

1.2.1.2. *Ciclo de vida de entrega completa*

Aunque el enfoque principal de DAD sea la entrega, se toman en cuenta todos los factores tienen influencia sobre ella. El ciclo de vida de DAD abarca al proyecto completamente, desde la concepción de la idea para el producto, pasando por la entrega, hasta las operaciones de soporte y mantenimiento. Durante cada etapa se pueden tener diversas iteraciones, en la Figura 2 se muestra un diagrama de alto nivel representando lo descrito:

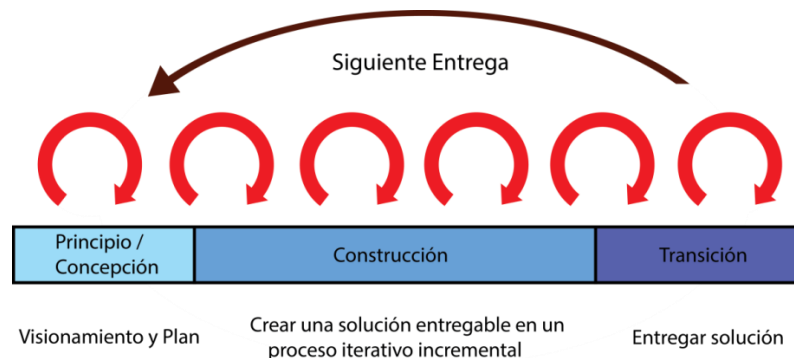


FIGURA 2 Ciclo de vida de entrega completa

Autor: Gustavo W. Moyano

1.2.1.3. Impulsado por Metas

DAD visualiza al proyecto como un conjunto de metas a cumplir. Por ejemplo, Scrum utiliza como práctica técnica por defecto un Product Backlog³ para la administración de requerimientos, mientras que DAD dice



FIGURA 3 Impulsado por Metas

Autor: Gustavo W. Moyano

que en la etapa de construcción del producto, un objetivo a cumplirse es direccionar los cambios de requerimientos de manera efectiva. Una forma de direccionar los cambios de requerimientos es, claro está, un Product Backlog,

pero no siempre calzará ésta técnica perfectamente en el presente proyecto. La visión de DAD por cumplimiento de objetivos le permite ser completamente flexible y adaptable a cualquier situación.

1.2.1.4. Conciencia Empresarial

Es uno de los aspectos más importantes que tiene el marco de trabajo DAD. Los equipos de trabajo DAD coexisten en el ecosistema empresarial, de tal manera que sus productos no tienen impacto alguno sobre el trabajo

³ Lista de características cumplidas/a cumplirse con una prioridad específica

de otros grupos, por otro lado, óptimamente sus soluciones apalancan la funcionalidad ya existente en la empresa. Las características visibles de los profesionales DAD son:

- Trabajo conjunto con personal de diferentes departamentos de la empresa, como arquitectos de sistema y administradores de cartera.
- Adaptabilidad y compromiso con la empresa.
- Apalancamiento de bienes empresariales, incluyendo sistemas y fuentes de datos existentes.
- Refactorización de bienes empresariales con el fin de mejorar el ecosistema organizacional.
- Adopción de una cultura DevOps⁴

La conciencia empresarial es importante por diversas razones. Primero, se puede reducir el tiempo y el costo total de entrega por medio del uso y provecho de implementaciones ya existentes. En otras palabras, equipos DAD pueden optimizar su tiempo al no tener que “reinventar la rueda”, y pueden concentrarse en dar el valor que el cliente realmente necesita. Segundo, los equipos DAD toman fácilmente la dirección correcta al trabajar con profesionales de la empresa. Y tercero, se incrementa la posibilidad de que el equipo de entrega aporte prolijidad al “todo” organizacional, y no solo la parte de la solución que supuestamente les concierne directamente.

1.2.2. Ciclos de Vida de Entrega Completamente Ágil

El ciclo de vida de DAD, en el más alto nivel de abstracción, posee tres etapas claramente diferenciables:

⁴ Metodología de desarrollo de software centrada en la comunicación, colaboración e integración entre desarrolladores de software y profesionales de operaciones de TI. Viene de la combinación de las palabras en inglés Development (desarrollo) y Operations (operaciones).

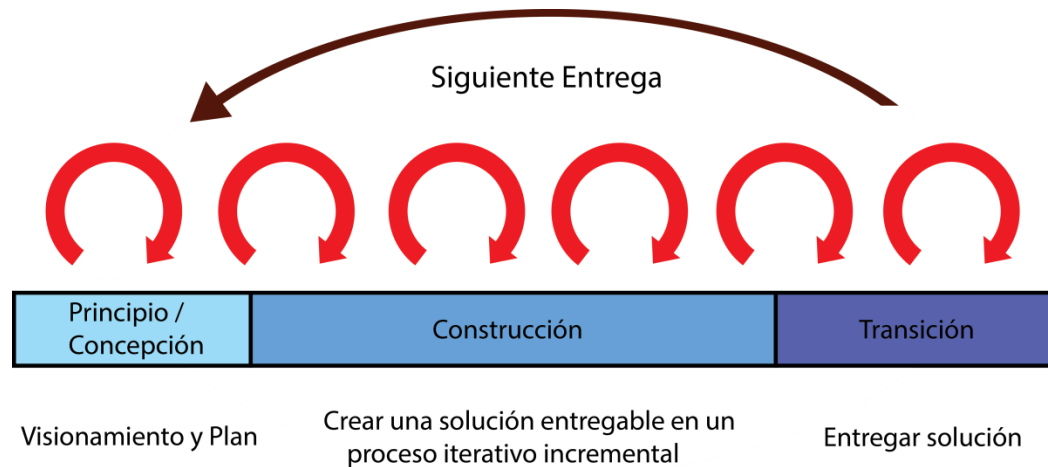


FIGURA 4 Ciclo de Vida de Entrega Completamente Ágil

Autor: Gustavo W. Moyano

- *Principio / Concepción*

Durante esta fase del proyecto se llevan a cabo las actividades de iniciación. Aunque la palabra “fase” tiende a sacudir a la comunidad ágil⁵, la mayoría de equipos realizan un conjunto de actividades al inicio del proyecto. Algunos equipos le dan el título de Iteración/Sprint 0⁶ a dicho conjunto, sin embargo, normalmente estas actividades toman más tiempo que una Iteración/Sprint (el “2013 Agile Project Initiation survey”⁷ encontró que usualmente un equipo ágil se toma un mes en la etapa de Principio / Concepción, mientras que la mayoría de Iteraciones / Sprints duran sólo 2 semanas). En la fase de principio o concepción del ciclo de vida de DAD se realiza un ligero conjunto de actividades que permiten enmarcar adecuadamente al proyecto. Se necesita disciplina para mantener esta fase corta.

- *Construcción*

⁵ Grupo de equipos que utilizan metodologías ágiles de desarrollo de software.

⁶ Inicio del ciclo de vida en la metodología Scrum.

⁷ Encuesta realizada en Julio 2013 con el objetivo de averiguar cuánto esfuerzo se toma un equipo ágil en la iniciación de su proyecto y qué actividades son realizadas.

Durante esta fase del ciclo, se producirá incrementalmente una solución entregable. La solución bien puede ser completada a través de una serie de iteraciones (Sprints en Scrum), o mediante un flujo continuo. La solución es desarrollada bajo un marco híbrido de prácticas: Scrum, XP, Agile Modeling, Agile Data, y otros métodos.

- *Transición*

La entrega de soluciones a los clientes es claramente una tarea compleja y sofisticada que DAD toma en consideración. Los equipos DAD y la empresa en general, dinamizarán sus procesos de despliegue haciendo cada vez más cortas las fases de transición, y en un marco ideal, desaparecen convirtiéndose en una actividad. Se requiere disciplina para lograr dicha conversión.

En un nivel más bajo de abstracción, se pueden observar dos etapas más en el ciclo de vida de DAD:

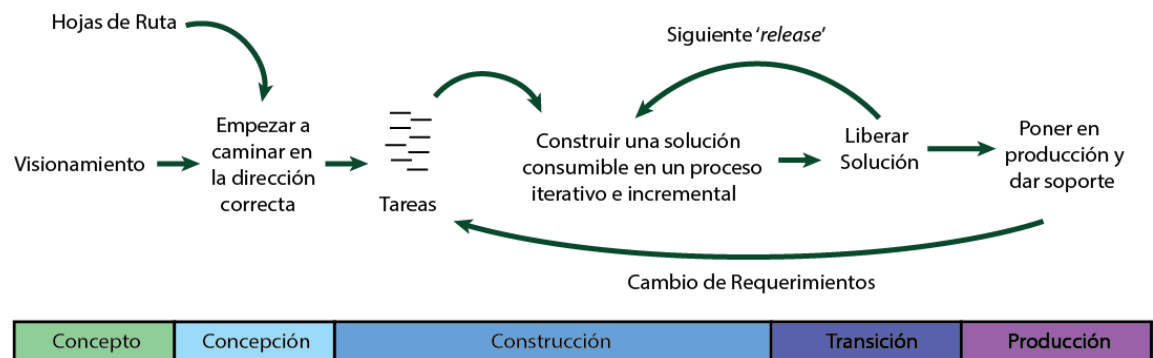


FIGURA 5 Ciclo de Vida de Entrega Completamente Ágil – Siguiendo nivel

Autores: Scott W Ambler - Disciplined Agile Consortium

Una etapa antes de la de concepción llamada Concepto. Hablando en términos empresariales, en ésta etapa se toman en cuenta proyectos potenciales y productos a ser definidos, identificados, priorizados y presentados a los inversores como propuesta, para posteriormente empezar una etapa de Concepción. Por otro lado, después de la Transición existe una etapa en la que la solución es puesta en producción

y el equipo se encarga de dar el soporte adecuado, ya sea para mantenerla o retirarla de producción.

Estas etapas permiten visualizar de manera macro la forma en que se lleva a cabo el desarrollo del proyecto, sin embargo, la metodología no prescribe y obliga a utilizar un ciclo de vida único, todo lo contrario, es totalmente flexible y se acopla a las necesidades de cada proyecto. A continuación se muestran cuatro ejemplos de ciclos de vida tomados para proyectos diferentes, desarrollados por equipos DAD:

1.2.2.1. *Ciclo de Vida DAD Ágil/Básico: Extendiendo a Scrum*

Éste ciclo de vida es básicamente una extensión del ciclo de construcción de Scrum, posee varias características interesantes:

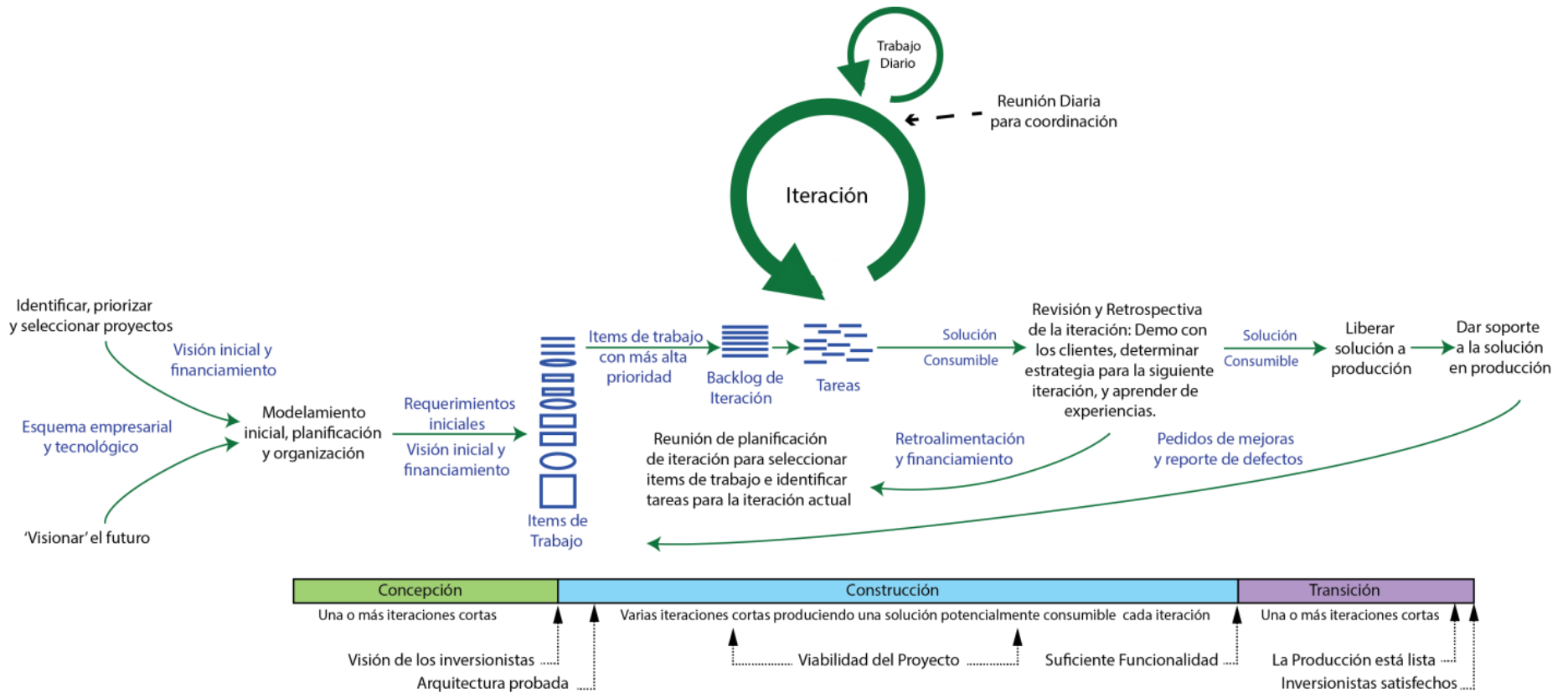


FIGURA 6 Ciclo de vida DAD Extendiendo a Scrum

Autores: Scott W Ambler - Disciplined Agile Consortium

- **Es basado en iteraciones.** Tal y como lo hacen otras metodologías ágiles, incluyendo Scrum y XP, la solución es construida incrementalmente en secciones de tiempo llamadas iteraciones (lo que Scrum llama 'Sprints').
- **Evita terminología Scrum.** Aunque el ciclo de vida se basa en el de Scrum, en DAD se utiliza lenguaje regular como terminología. Como por ejemplo los 'Sprints' de Scrum son llamados iteraciones. El lenguaje realmente es irrelevante, así que si un equipo se siente cómodo utilizando terminología Scrum, puede sentirse libre de hacerlo.
- **También muestra entradas que están fuera del ciclo de entrega.** El diagrama anterior mostró el ciclo de entrega, la siguiente figura nos mostrará que existen actividades que se llevan a cabo.
- **Se maneja una lista de *work items*, no un *product backlog*.** DAD tiene un enfoque más amplio que Scrum y necesita de técnicas robustas para la administración del cambio. Dentro de la lista de *work items* se incluyen defectos y requerimientos funcionales, así como también requerimientos no funcionales como entrenamiento, vacaciones y asesoría a otros equipos. La priorización dada abarca a todas las actividades a realizarse y no solo a las que giran en torno a los requerimientos.
- **Incluye metas expresadas explícitamente.** Al final del diagrama se encuentran unas pocas ligeras metas que cada equipo DAD debe luchar por conseguir. Estas metas son un aspecto importante en la gobernancia del proyecto en general.

1.2.2.2. *Ciclo de Vida DAD Avanzado/Ajustado*

Existen varias características interesantes de éste ciclo de vida:

- **Soporta un flujo de entrega continua.** En este ciclo de vida, el desarrollo de la solución se lleva a cabo cuando, y al paso que la solución lo necesite. El equipo encuentra trabajo en su camino y es realizado cuando exista la capacidad, no necesariamente en el pulso regular de una iteración.
- **Las prácticas/técnicas tienen sus propias cadencias.** Con un sistema de iteraciones o *sprints*, muchas prácticas (como planificación detallada, retrospectivas, demos, modelamiento detallado, entre otras) son

efectivamente llevadas a cabo dentro del mismo tiempo de una iteración. Mientras que con un enfoque avanzado/ajustado, las prácticas son realizadas cuando amerite, cuando tenga sentido desarrollarlas, mas no porque el calendario lo dicta.

- **Tiene un *work ítem pool*⁸.** No todas las actividades de trabajo son creadas de la misma manera. Aunque las técnicas escogidas para priorizarlas sean de manera estándar (mediante un *product backlog*) o por valor ofrecido (como DAD o RUP sugieren), aparecerán actividades que deben inevitablemente ser aceleradas, como por ejemplo corregir un problema severo en la producción o también aparecerán actividades que dependan de una fecha determinada, como por ejemplo actividades resultantes de la legislación. Un *work ítem pool* permitirá al ciclo de vida acoplarse a la realidad de una manera más cómoda que una simple lista con prioridades. Dividirá el trabajo en grupos de actividades, exigiendo cada grupo un enfoque diferente de priorización.

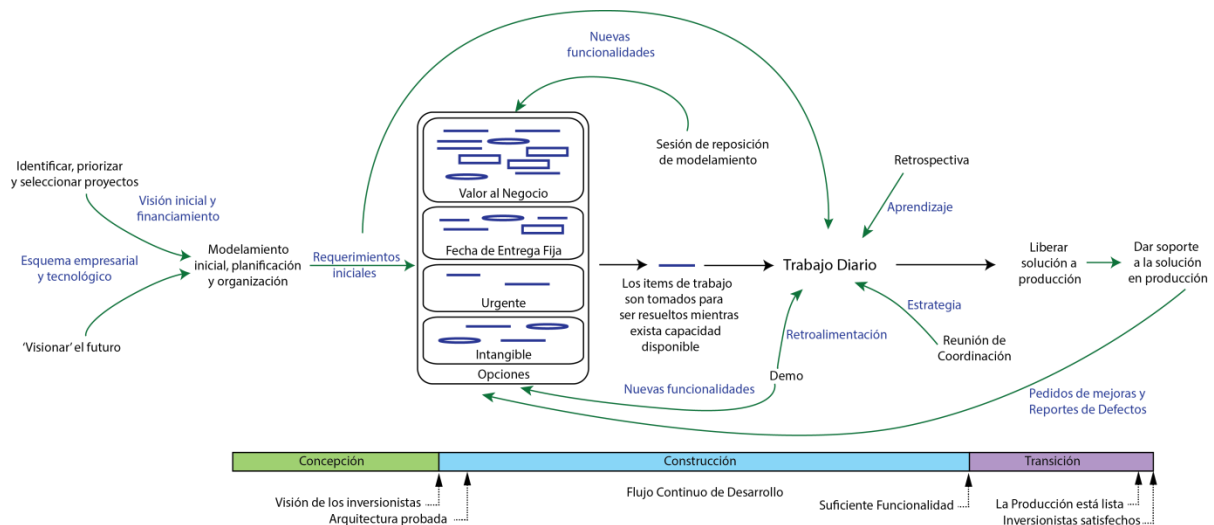


FIGURA 7 Ciclo de vida DAD Avanzado / Ajustado

Autores: Scott W Ambler - Disciplined Agile Consortium

⁸ Es una lista de ítems de trabajo, cada ítem tiene una prioridad distinta, lo que permite que en cada iteración se tomen en cuenta los ítems con mayor prioridad. Pueden ser historias de usuario, acciones o decisiones a tomarse fuera o dentro del sistema e incluso requerimientos no funcionales.

Este ciclo es llamado Avanzado/Ajustado porque es el resultante de equipos con experiencia. Usualmente los equipos inician con el anterior ciclo de vida, pero mientras se aprende, se trabaja y se gana experiencia, el ciclo de vida se vuelve más “ajustable”.

1.2.2.3. Ciclo de Vida DAD de Entrega Continua

Éste ciclo de vida es básicamente una versión aún más ajustable del anterior. El producto es creado, desarrollado y entregado al mercado de una manera constante, diariamente, por semana o por mes.

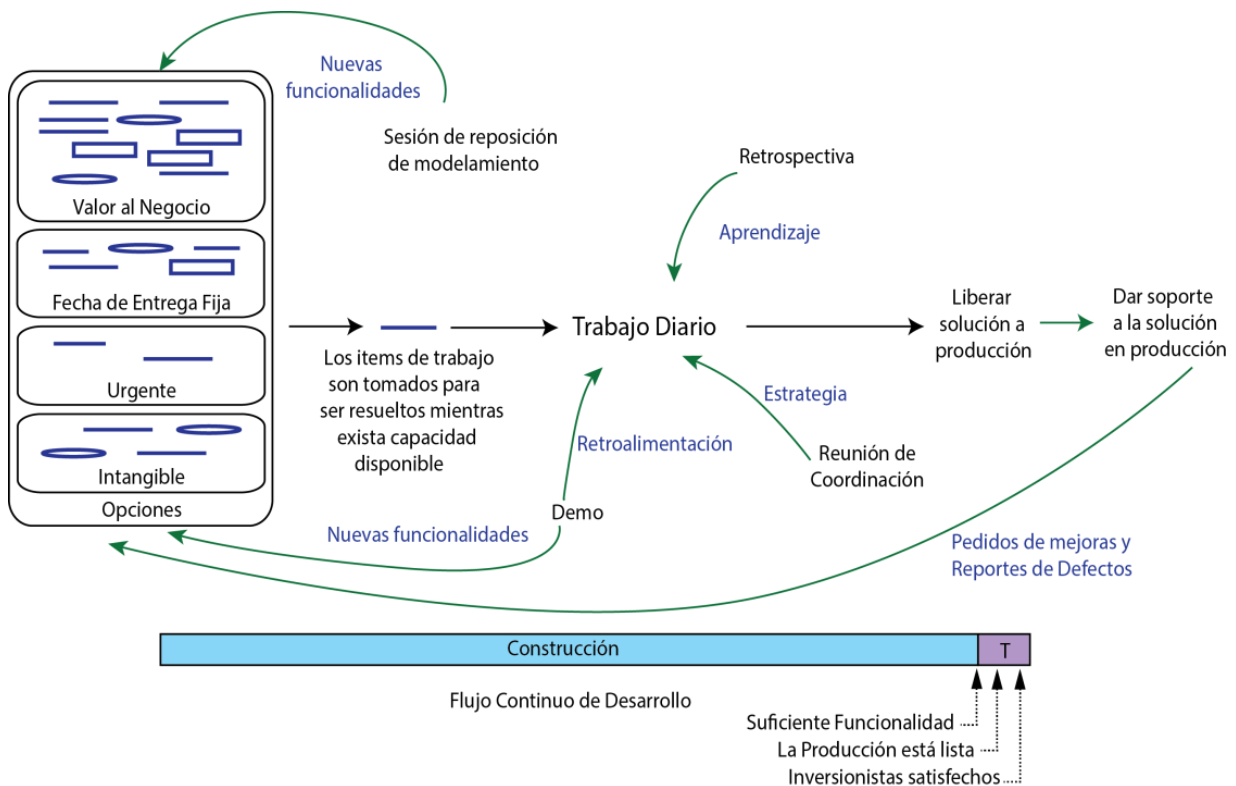


FIGURA 8 Ciclo de vida DAD Entrega Continua

Autores: Scott W Ambler - Disciplined Agile Consortium

1.2.2.4. Ciclo de Vida DAD Exploratorio

Éste ciclo de vida es seguido por equipos que se encuentran iniciando un nuevo proyecto, en una etapa de investigación. Las bases de desarrollo del

producto son claras, pero las especificaciones de qué necesita el cliente, la demanda puntual del mercado, no es conocida al cien por ciento.

El ciclo de vida exploratorio, por sus características, puede encajar como fase de principio/concepción para otros ciclos de vida.

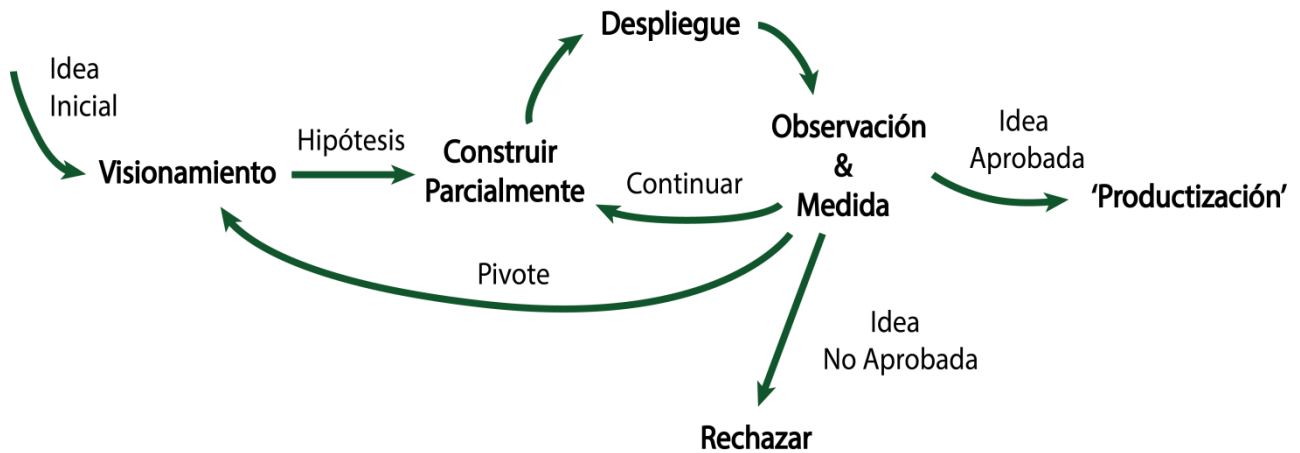


FIGURA 9 Ciclo de vida DAD Exploratorio

Autores: Scott W Ambler - Disciplined Agile Consortium

Existen seis actividades principales:

- **Visionamiento.** El equipo explora la idea e identifica una estrategia de implementación potencial para la misma. Se necesita construir una hipótesis lo más viable para saber la demanda real del cliente.
- **Construcción parcial.** El equipo debe invertir el esfuerzo suficiente para construir una solución que pruebe la hipótesis. Esto se conoce como el Producto mínimamente viable (PMV).
- **Despliegue.** Debe ser realizado una vez que la solución actual esté lista para ser desplegada en el entorno en el que se pueda probar la hipótesis.
- **Observación y Medida.** Una vez que la solución está disponible para la producción, se necesita determinar qué aspectos son de interés para el grupo de clientes. Para lograr este objetivo se necesita recoger datos a través de instrumentos adheridos al producto. Con la información construida a partir de dichos datos se puede tomar una decisión, si la funcionalidad es bien recibida, el siguiente paso será continuar con la estrategia y añadir más funcionalidades, incluso la funcionalidad añadida

puede llegar a ser óptima para llegar a formar parte del producto final. Si la funcionalidad no fue bien recibida, el equipo decidirá refinar la idea y generar una nueva hipótesis, o cancelar la funcionalidad completamente.

- **Rechazar.** Existirán ciertas ocasiones en las que se descubre que la idea del producto no promete resultados. La ventaja es que si una idea es evidentemente improductiva, la energía planeada para su desarrollo será efectivamente canalizada hacia otras estrategias.
- **“Productización”.** Después de varias iteraciones de construcción parcial, despliegue y observación y medida, se logra identificar de manera clara un producto que tendrá éxito en el mercado. El equipo puede escoger continuar con el ciclo exploratorio para su producción, pero usualmente otros ciclos de vida son escogidos, y este ciclo es tomado como la fase de principio / concepción.

1.2.3. Enfoque guiado por Metas - Metas de Proceso

El marco de trabajo DAD se lleva a cabo mediante un desarrollo guiado por metas u objetivos, con el propósito de tenerlos como guías para todas las personas involucradas en el proyecto. Dichos objetivos ayudan en la toma de decisiones relacionadas con los procesos de negocio, para entallar y escalar estrategias de desarrollo ágil, de tal manera que se personalizan soluciones dentro del contexto de la situación.

Teniendo en mente que cada proyecto se encuentra en una situación única, que cada equipo de desarrollo es diferente de otro y que cada miembro de un equipo posee distintos talentos y habilidades que otro, se debe encontrar la manera efectiva de llevar a cabo la solución. La estrategia sugerida por el marco de trabajo DAD, el desarrollo guiado por metas, es un enfoque que provee de varias opciones que un equipo puede utilizar para personalizar el marco de trabajo para su situación.

La figura 10 resume las metas de proceso para toma de decisiones sugeridas por el marco de trabajo DAD. Existen en total 22 metas de proceso, cada una es explicada en un diagrama de meta de proceso. Un equipo de desarrollo ágil decidirá cómo aplicar cada meta de proceso, de manera que se refleje la situación en la que se encuentren.

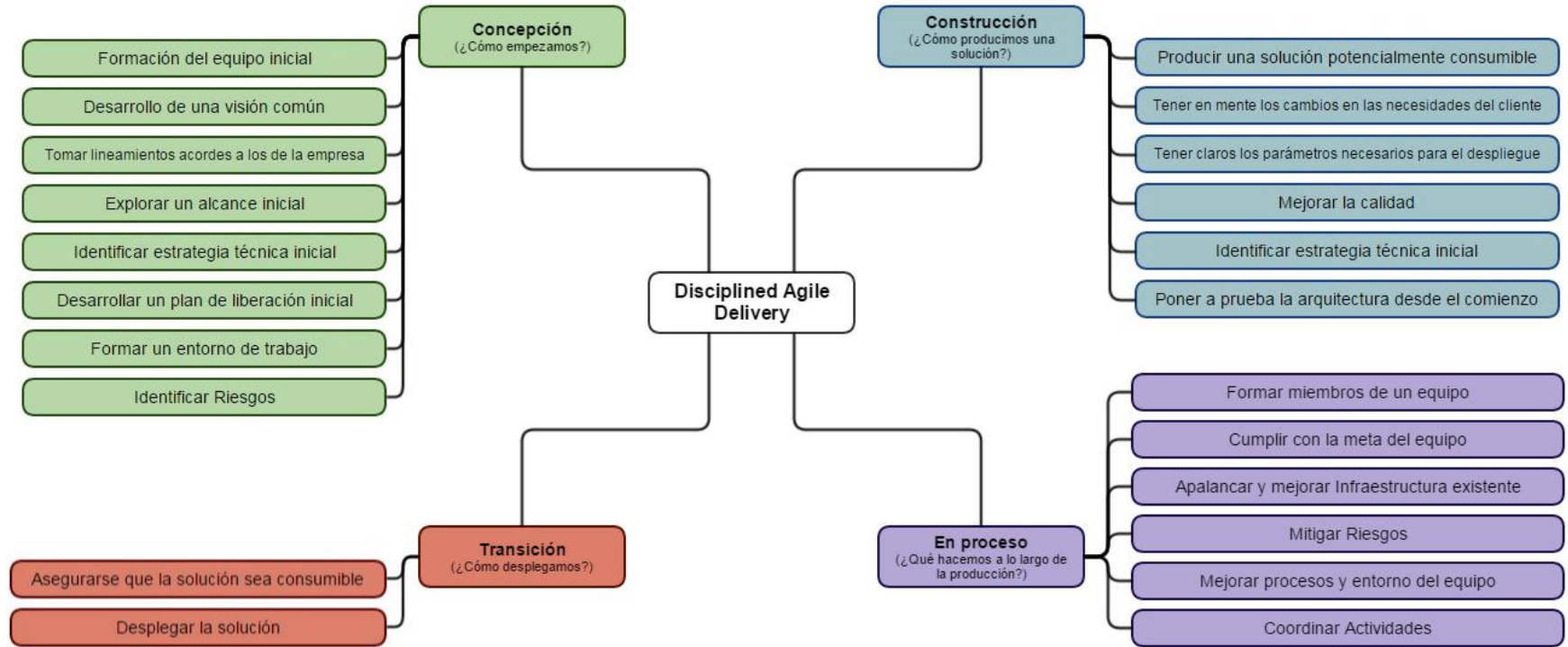


FIGURA 10 Metas de Proceso

Autores: Scott W Ambler - Disciplined Agile Consortium

1.2.3.1. Diagramas de Metas de Proceso

En los diagramas de Metas de Proceso se muestran las opciones que tiene cada meta para que los equipos ágiles las puedan seguir, y cada una con diferentes técnicas de aplicación. En la FIGURA 11 se muestra un ejemplo de diagrama:

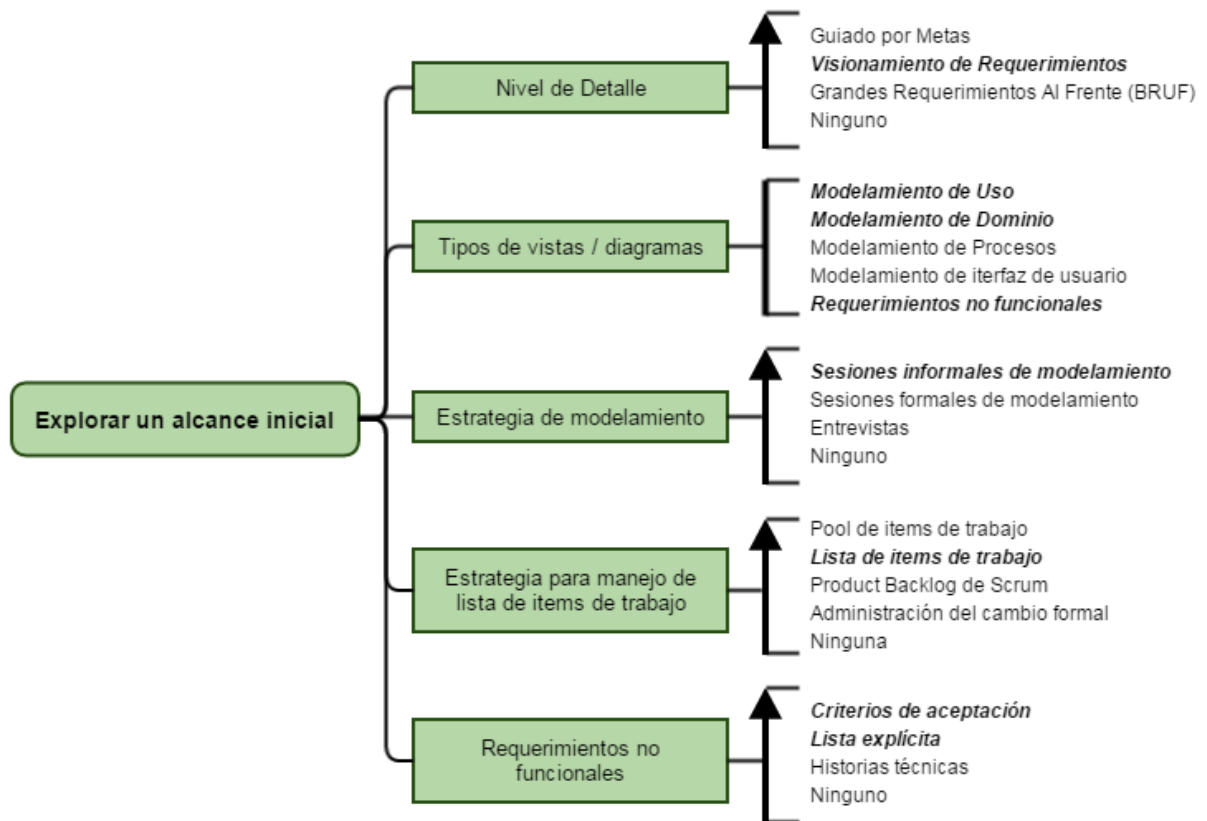


FIGURA 11 Ejemplo de diagrama de Meta de Proceso

Autores: Scott W Ambler - Disciplined Agile Consortium

Se pueden destacar algunos puntos importantes con respecto a los diagramas de Metas de Procesos:

- **Existen aún más opciones.** Aunque los diagramas proveen de una buena representación de las opciones que se pueden tomar, siempre existen más estrategias y prácticas que surgen a diario.

- **Algunas opciones en realidad son categorías de opciones.** Tomando como ejemplo la figura, se puede ver que en los tipos de diagramas se tiene modelamiento de uso, modelamiento de dominio, etc. Existen varias formas de hacer modelamiento de uso, como historias de usuario, casos de uso, escenarios de uso, así como también para modelamiento de dominio como por ejemplo: modelamiento de datos, modelamiento conceptual, diagrama de clases UML, etc.
- **Cada opción puede ser cambiada.** No existe el concepto de “la mejor práctica”, cada opción puede ser mejor o peor en cada situación y escenario diferentes.
- **Algunas opciones son generalmente mejores que otras.** Cuando se muestre una flecha en la parte izquierda de la lista de opciones, es un indicador que las opciones más cercanas al tope de flecha son las comúnmente consideradas más efectivas desde un punto de vista de desarrollo ágil que las que están al final.
- **Opciones por defecto son resaltadas.** Para facilitar la utilización de los diagramas, se resaltan (con fuente negrita e itálica) las opciones que generalmente son utilizadas por equipos con una trayectoria corta o nula en el desarrollo ágil.

1.2.4. Conciencia Empresarial

La conciencia empresarial es uno de los aspectos claves del marco de trabajo DAD. La observación es que los equipos DAD trabajan dentro del ecosistema empresarial de la compañía conjuntamente con los otros equipos de trabajo, otros sistemas ya implementados no deberían verse afectados por la implementación del producto desarrollado por el equipo DAD. La nueva solución, por otro lado, debería apalancar la funcionalidad y la información ya disponibles en la producción actual. Dentro de las funciones habituales de la compañía, los equipos que cumplen distintos objetivos trabajaran en paralelo, entre ellos el equipo DAD. La visión en el campo es aprovechar del trabajo que otros equipos ya han llevado a cabo y viceversa, la conciencia empresarial de los equipos debe permitir cumplir con esta visión.

En este artículo se describirá lo siguiente:

- Cinco niveles de conciencia
- Significado de la conciencia empresarial
- Por qué la conciencia empresarial es importante
- Desafíos para la conciencia empresarial

1.2.4.1. Cinco niveles de conciencia

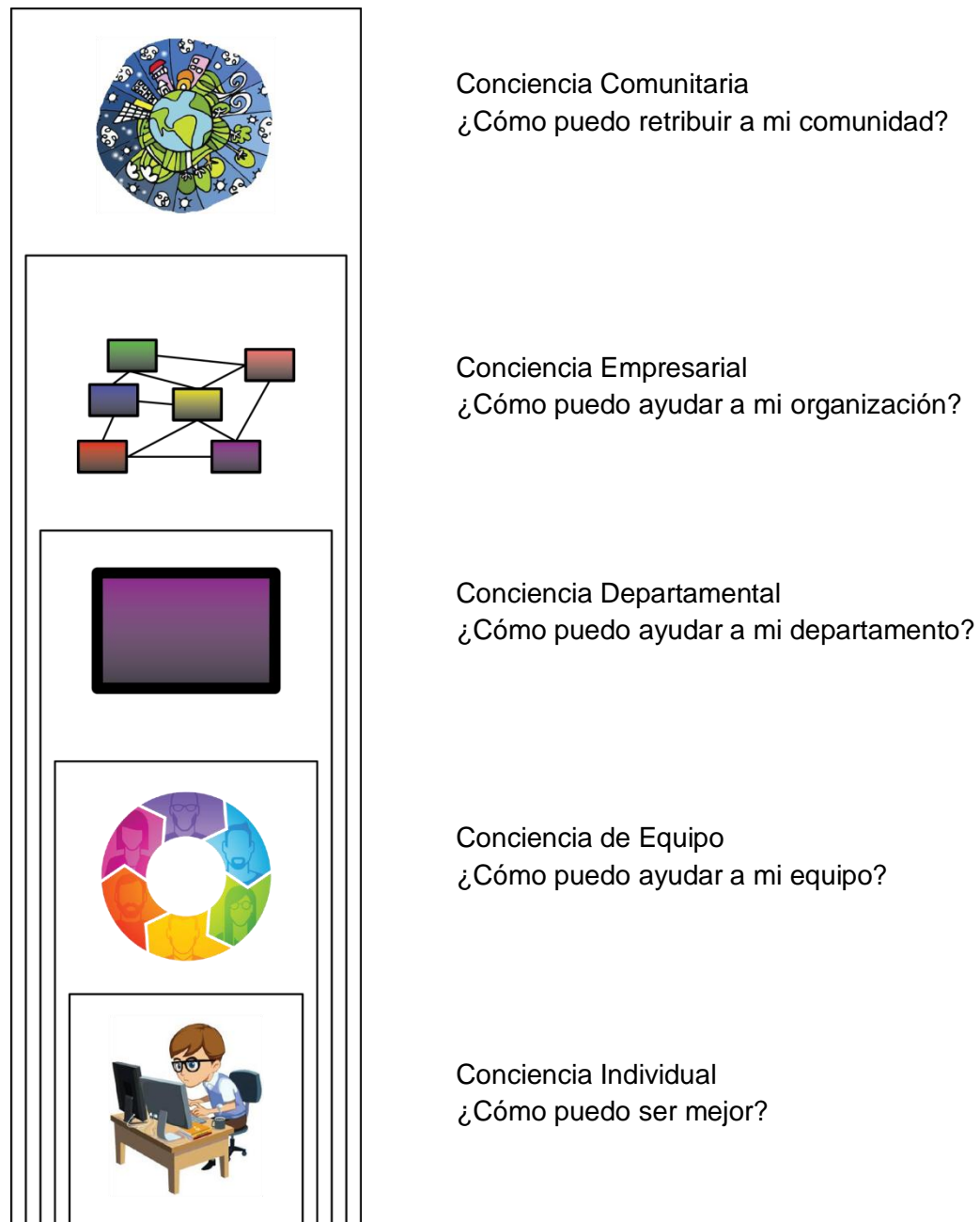


FIGURA 12 Cinco niveles de Conciencia

Autor: Gustavo W Moyano

1.2.4.1.1. Conciencia Individual

La conciencia individual se enfoca en cómo alguien puede mejorar mediante herramientas como aprender nuevas habilidades, agudizar sus sentidos, aumentar su perspicacia, ganar experiencia, etc.

1.2.4.1.2. Conciencia de Equipo

El enfoque de la conciencia de equipo es que los miembros del equipo aprenden y mejoran juntos. Esta filosofía ha formado parte de la “comunidad ágil” por algún tiempo, la mayoría de veces para beneficio, pero en algunas también para detrimento. Si los miembros del equipo se enfocan en el verdadero trabajo en equipo, es decir tienen conciencia de equipo, su productividad incrementará. Por otro lado, si los esfuerzos del equipo no están alineados a las metas de la organización, inmediatamente aparecerán disfunciones.

1.2.4.1.3. Conciencia Departamental

La gente toma en consideración las necesidades de su departamento, no solo de su equipo. En este caso, los desarrolladores se concentran en mejorar el proceso de TI de manera global, probablemente adoptando una cultura DevOps en lugar de una mentalidad tradicional.

1.2.4.1.4. Conciencia Empresarial

Las personas son motivadas a velar de manera global por las necesidades de toda la compañía, asegurando así la contribución positiva de sus acciones hacia los objetivos de la empresa y no solo a las metas sub óptimas del equipo.

1.2.4.1.5. Conciencia Comunitaria

La gente tiene en cuenta las necesidades de la comunidad, intentando por los medios que puedan entregar de vuelta a los demás compartiendo su conocimiento, esforzándose por aprender por sí mismos, y esforzándose por ayudar a otros, quienes no necesariamente trabajen dentro de la empresa, e incluso a quienes tal vez ni siquiera conozcan. La Conciencia Comunitaria es un aspecto clave en certificaciones Disciplined Agile, como Black o Green Belt⁹.

⁹ Certificaciones DAD

1.2.4.2. *Significado de la conciencia empresarial*

La Conciencia Empresarial es un aspecto importante de la auto disciplina, porque uno como profesional debe esforzarse por hacer lo que es correcto para su organización y no sólo lo que es interesante para uno mismo. Los equipos de desarrollo de manera aislada pueden optar por construir algo desde cero, utilizar diferentes herramientas de desarrollo o crear diferentes fuentes de datos, incluso cuando dentro del esquema de la compañía ya existan y estén desarrolladas soluciones buenas y estables, que se han instalado con éxito, y han sido evaluadas y configuradas. Los profesionales disciplinados y ágiles, por otro lado:

- Trabajarán en estrecha colaboración con los profesionales de la empresa. Esto incluye trabajar en estrecha colaboración con arquitectos técnicos y reutilizar ingenieros de la empresa con el fin de aprovechar y mejorar los existentes y "ser" infraestructura técnica; trabajar con arquitectos de negocios empresariales y gestores de cartera para adaptarse al ecosistema de negocios en general; con los altos directivos que deben regir los diferentes equipos apropiadamente; con el personal de operaciones para apoyar las operaciones (DevOps) los esfuerzos de desarrollo global de su organización y; con administradores de datos para que accedan y mejoren fuentes de datos existentes; con personas de apoyo de desarrollo de TI para comprender y seguir la orientación empresarial de TI; y con expertos en negocios que comparten sus conocimientos del mercado, previsiones de ventas, previsiones de servicios y otros asuntos importantes. En otras palabras, los equipos DAD deben adoptar un modo de pensar "globalmente empresarial".

1.2.5. Soluciones “Consumibles”

Un aspecto importante de la adopción de la metodología DAD es la evolución del pensamiento del desarrollador. La filosofía debe dar un giro de tal magnitud, que se deje a un lado el pensar que simplemente se está produciendo software, a tener en mente que se provee al cliente con una solución consumible, que cumple con cada una de sus necesidades, no sólo dentro de la organización, sino también en el entorno económico, cultural y

técnico en el que se lo maneje. La observación fundamental es que, como profesionales de IT, abarcamos mucho más que el desarrollo de software.

1.2.6. El Manifiesto Ágil Disciplinado (“The Disciplined Agile Manifesto”)

El Manifiesto Ágil Disciplinado es una extensión del original “Manifiesto para desarrollar Software Ágil” (escrito en 2001), en el que se reflejan las filosofías detrás del marco de trabajo DAD.

Se Valoran:

Individuos e interacciones por sobre procesos y herramientas,

Soluciones consumibles por sobre documentación comprensible,

Colaboración de stakeholders por sobre negociación por contrato,

Respuesta al cambio por sobre seguir el plan al pie de la letra.

Cuando hay valor en los ítems de la derecha, los seguidores de la filosofía ágil disciplinada valoran los ítems de la izquierda.

Principios:

- La prioridad es satisfacer a los interesados mediante la entrega rápida y continua de soluciones de alto valor.
- Se da la bienvenida a cambios en los requisitos, incluso al final del ciclo de vida de la entrega de la solución. Los procesos ágiles aprovechan el cambio para dar ventaja competitiva al cliente.
- Se entregan soluciones consumibles con frecuencia, entre 2 semanas y 2 meses, prefiriendo acortar la escala de tiempo.
- Los clientes y los desarrolladores trabajan juntos cada día durante el proyecto.
- Se construyen equipos con personas motivadas. Trabajan bajo un entorno y con el soporte que necesitan. Existe confianza en ellos para hacer el trabajo.

- El método más eficiente y efectivo para comunicar información al equipo, y dentro de éste es la conversación cara a cara.
- Las soluciones consumibles son la principal medida del avance.
- Los procesos ágiles promueven la entrega sostenible. Los clientes, desarrolladores y usuarios deberían poder mantener un ritmo constante indefinidamente.
- La atención continua a la excelencia técnica y al buen diseño fundamenta la agilidad.
- Simplicidad (el arte de maximizar la cantidad de trabajo todavía no realizado) es esencial.
- Las mejores arquitecturas, requerimientos y diseños emergen de equipos con capacidad auto-organizativa.
- El equipo reflexiona como ser más efectivo a intervalos regulares, como consiguiente, afina y reajusta su funcionamiento.
- Existe un apalancamiento para la evolución de la productividad de los bienes de la empresa, con un trabajo conjunto con el equipo de desarrollo y la gente responsable de los bienes.
- El equipo visualiza el flujo de trabajo y lo mantiene equilibrado, logrando así un flujo bajo poca presión (el trabajo en proceso se mantiene al mínimo con tareas en paralelo).
- El ecosistema organizacional debe evolucionar de tal manera que se pueda reflejar y resaltar los esfuerzos dentro de equipos ágiles, y aun así, ser lo suficientemente flexible para poder dar soporte a otros equipos no ágiles o híbridos.

1.2.7. Ventajas y Desventajas de la Metodología

1.2.7.1. *Ventajas*

- Es un marco de trabajo híbrido basado en prácticas ágiles y ‘esbeltas’.
- La visión orientada a metas u objetivos a cumplir (hitos), permite entender la realidad verdadera de los ciclos de vida de entrega ágil a gran escala.
- Posee flexibilidad que enfatiza el entalle de la adopción de prácticas ágiles.
- Posee una arquitectura técnica fuerte, se hace foco a la ingeniería.
- Existe claridad en la gobernabilidad de la administración de programas y productos.

1.2.7.2. *Desventajas*

- El nicho de mercado en el que se pueda adoptar la metodología es desconocido
- El proceso de entrenamiento y certificación y el refinamiento de sus conocimientos base son limitados.
- Es difícil para una empresa decidir cuándo utilizar DAD, por lo que se necesita inevitablemente consultorías agilistas¹⁰.

¹⁰ Que aplica metodologías ágiles de desarrollo

1.3. Herramientas de Desarrollo

1.3.1. Java Persistence Application Programming Interface (JPA)

Framework Java que administra datos relacionales en aplicaciones usando la Plataforma Java SE y Java EE.

JPA en este contexto cubre tres áreas:

- La API en sí misma, definida en el paquete 'javax.persistence'
- El lenguaje de consulta Java Persistence Query Language (JPQL).
- Metadatos objeto/relacional.

El objetivo que persigue el diseño de esta API es no perder las ventajas de la orientación a objetos al interactuar con una base de datos (siguiendo el patrón de mapeo objeto-relacional), como sí pasaba con EJB2, y permitir usar objetos regulares (conocidos como POJOs).

Para éste caso, en el servidor de aplicaciones (Jboss) se utilizará la herramienta que viene embebida dentro del mismo, Hibernate.

1.3.2. Hibernate

Hibernate¹¹ es una herramienta de Mapeo objeto-relacional (ORM)¹² para la plataforma Java (y disponible también para .Net con el nombre de NHibernate) que facilita el mapeo de atributos entre una base de datos relacional tradicional y el modelo de objetos de una aplicación, mediante archivos declarativos (XML¹³) o anotaciones en los beans de las entidades que permiten establecer estas relaciones.

Hibernate es software libre, distribuido bajo los términos de la licencia GNU LGPL¹⁴.

¹¹ Definición tomada

¹² Técnica de programación para convertir datos entre el sistema de tipos utilizado en un lenguaje de programación orientado a objetos y la utilización de una base de datos relacional como motor de persistencia.

¹³ eXtensible Markup Language (lenguaje de marcas extensible)

¹⁴ GNU Lesser General Public License / Licencia de Software en la que cualquier persona tiene permisos de copia y distribución

La característica principal de Hibernate es su mapeo desde clases Java hacia tablas de bases de datos (y desde tipos de dato Java a tipos de dato SQL). Hibernate facilita también un mecanismo de consulta y manejo de datos. Hibernate genera llamadas SQL, y ahorra el trabajo de conversión de resultados a objetos.

1.3.3. Java RESTful Services

La Transferencia de Estado Representacional (Representational State Transfer) o REST¹⁵ es una técnica de arquitectura de software para sistemas web.

En la actualidad, el término REST se usa para describir cualquier interfaz web simple que utiliza XML y HTTP, sin incluir protocolos de mensajería SOAP (Simple Object Access Protocol). Los sistemas de servicios web pueden ser contruidos de dos maneras:

De acuerdo con el Estilo Arquitectural REST de *Fielding*

Basados en interfaces XMLHTTP de acuerdo con el estilo de llamada a procedimiento remoto (RPC), pero sin usar SOAP.

Los sistemas que siguen los principios REST se llaman con frecuencia RESTful.

REST afirma que la web ha disfrutado de escalabilidad como resultado de una serie de diseños fundamentales clave:

Un **protocolo cliente/servidor sin estado**: cada mensaje HTTP contiene toda la información necesaria para comprender la petición. Como resultado, ni el cliente ni el servidor necesitan recordar ningún estado de las comunicaciones entre mensajes. Sin embargo, en la práctica, muchas aplicaciones basadas en HTTP utilizan cookies y otros mecanismos para mantener el estado de la sesión (algunas de estas prácticas, como la reescritura de URLs, no son permitidas por REST)

Un conjunto de **operaciones bien definidas** que se aplican a todos los recursos de información: HTTP en sí define un conjunto pequeño de

¹⁵ Definición tomada de "Representational state transfer" - Wikipedia

operaciones, las más importantes son **POST**, **GET**, **PUT** y **DELETE**. Con frecuencia estas operaciones se equiparan a las operaciones CRUD en bases de datos (ABMC en castellano: Actualizar, Borrar, Modificación y Consulta) que se requieren para la persistencia de datos, aunque POST no encaja exactamente en este esquema.

Una **sintaxis universal** para identificar los recursos. En un sistema REST, cada recurso es direccionable únicamente a través de su URI.

El **uso de hipermedios**, tanto para la información de la aplicación como para las transiciones de estado de la aplicación: la representación de este estado en un sistema REST son **típicamente HTML o XML**. Como resultado de esto, es posible navegar de un recurso REST a muchos otros, simplemente siguiendo enlaces sin requerir el uso de registros u otra infraestructura adicional.

1.3.4. Javascript

JavaScript ¹⁶(JS) es un lenguaje de programación interpretado, orientado a objetos, basado en prototipos, imperativo y dinámico.

Se utiliza principalmente al lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas. Existe también una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al lenguaje C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al

¹⁶ Definición tomada de "Javascript" - Wikipedia

lenguaje JavaScript de una implementación del Document Object Model (DOM¹⁷).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

Una cuarta edición está en desarrollo e incluirá nuevas características tales como paquetes, espacio de nombres y definición explícita de clases

1.3.5. Angular JS

Es un framework código abierto de JavaScript¹⁸, mantenido por Google, que ayuda con la gestión de aplicaciones SPA (Single Page Applications – aplicaciones de una sola página). Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (MVC), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos JSON estáticas o dinámicas.

Los objetivos de diseño:

- Disociar la manipulación del DOM de la lógica de la aplicación. Esto mejora la capacidad de prueba del código.
- Considerar a las pruebas de la aplicación como iguales en importancia a la escritura de la aplicación. La dificultad de las pruebas se ve reducida dramáticamente por la forma en que el código está estructurado.

¹⁷ Interfaz que proporciona un conjunto estándar de objetos para representar documentos html y xml.

¹⁸ Definición tomada de “AngularJS” – angularjs.org

- Disociar el lado del cliente de una aplicación del lado del servidor. Esto permite que el trabajo de desarrollo avance en paralelo, y permite la reutilización de ambos lados.
- Guiar a los desarrolladores a través de todo el camino de la construcción de una aplicación: desde el diseño de la interfaz de usuario, a través de la escritura de la lógica del negocio, hasta las pruebas.

1.3.6. HTML5

HyperText Markup Language 5 ¹⁹ es la quinta versión del lenguaje HTML, éste especifica dos variantes de sintaxis: un “clásico” HTML (text/html) y una variante XHTML que deberá ser servida como XML. Esta es la primera vez que HTML y XHTML se han desarrollado en paralelo.

Al no ser reconocido en viejas versiones de navegadores por sus nuevas etiquetas, se recomienda al usuario común actualizar a la versión más nueva, para poder disfrutar de todo el potencial que provee HTML5.

El desarrollo de este lenguaje de marcado es regulado por el Consorcio W3C.

HTML5 establece una serie de nuevos elementos y atributos que reflejan el uso típico de los sitios web modernos. Algunos de ellos son técnicamente similares a las etiquetas <div> y , pero tienen un significado semántico, como por ejemplo <nav> (bloque de navegación del sitio web) y <footer>. Otros elementos proporcionan nuevas funcionalidades a través de una interfaz estandarizada, como los elementos <audio> y <video>. Mejora el elemento <canvas>, capaz de renderizar elementos 3D en los navegadores más importantes (Firefox, Chrome, Opera, Safari e Internet Explorer).

Algunos elementos de HTML 4.01 han quedado obsoletos, incluyendo elementos puramente de presentación, como y <center>, cuyos efectos son manejados por Hojas de estilo en cascada. También hay un renovado énfasis en la importancia del scripting DOM para el comportamiento de la web 2.0.

¹⁹ Definición tomada de “HTML5 Tutorial” – w3schools.com

1.3.7. CSS 3

Hoja de estilo en cascada o CSS ²⁰(siglas en inglés de cascading style sheets) es un lenguaje usado para definir y crear la presentación de un documento estructurado escrito en HTML o XML2 (y por extensión en XHTML). El World Wide Web Consortium (W3C) es el encargado de formular la especificación de las hojas de estilo que servirán de estándar para los agentes de usuario o navegadores.

La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación.

La información de estilo puede ser definida en un documento separado o en el mismo documento HTML. En este último caso podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

A diferencia de CSS2, que fue una gran especificación que definía varias funcionalidades, CSS3 está dividida en varios documentos separados, llamados "módulos".

Cada módulo añade nuevas funcionalidades a las definidas en CSS2, de manera que se preservan las anteriores para mantener la compatibilidad.

Debido a la modularización del CSS3, diferentes módulos pueden encontrarse en diferentes estados de su desarrollo, de forma que a fechas de noviembre de 2011, hay alrededor de cincuenta módulos publicados, tres de ellos se convirtieron en recomendaciones oficiales de la W3C en 2011: "Selectores", "Espacios de nombres" y "Color".

Algunos módulos, como "Fondos y colores", "Consultas de medios" o "Diseños multicolumna" están en fase de "candidatos", y considerados como razonablemente estables, a finales de 2011, y sus implementaciones en los diferentes navegadores son señaladas con los prefijos del motor del mismo.

²⁰ Definición tomada de "Cascading Style Sheets" - Wikipedia

2. CAPÍTULO 2 - CASO DE ESTUDIO

2.1. Giro del Negocio

2.1.1. Datos de la Organización

"Bethania" es un centro creado para responder a las necesidades de formación, reflexión y descanso. Está dirigido a organizaciones religiosas y otras instituciones para la realización de convenciones, cursos de formación, seminarios, talleres, convivencias.

"Bethania" cuenta con un equipo de colaboradores, bajo la responsabilidad de las Hermanas "Salesias", cuyo compromiso es brindar comodidad y servir a sus invitados.

Los amplios jardines y espacios verdes le permiten estar en contacto con la naturaleza, las instalaciones están arquitectónicamente diseñadas para acoger y facilitar la interacción de grupos y el descanso de los participantes en habitaciones confortables.

Los beneficios de "Bethania" son:

- Un auditorio multimedia con capacidad para 60 personas.
- Dos salas múltiples con capacidad para 60 personas.
- Tres salas de reuniones con diseño ergonómico con capacidad para 20 personas.
- 56 habitaciones individuales con baño privado.
- 23 habitaciones dobles con baño privado.
- Una oficina de secretaría, exclusiva para los eventos.
- Un counter de recepción.
- Dos salas de espera.
- Una capilla para 200 personas.
- Un oratorio para 50 personas.
- Servicio de alimentación.

2.1.2. Misión

Bethania es un centro de convenciones especializado en organizar eventos de capacitación, formación, reflexión y descanso, dirigido a organizaciones religiosas y otras instituciones, brindando servicios de calidad bajo la responsabilidad de las Hermanas “Salesias”.

2.1.3. Visión

Bethania se proyecta como una organización innovadora para brindar comodidad y el mejor servicio a sus invitados y clientes a costos razonables, brindando confianza, seguridad y con la certeza de que todos los clientes queden satisfechos.

2.1.4. Objetivos

- Organizar eventos de capacitación, formación, reflexión y descanso en forma personalizada de acuerdo a los requerimientos de los clientes.
- Facilitar a los clientes la selección de los servicios requeridos.
- Brindar confianza y seguridad a sus clientes.
- Contribuir con máximo esfuerzo de trabajo para garantizar el mejor servicio.
- Trabajar con responsabilidad y vocación de servicio.
- Cumplir con los compromisos contraídos.
- Destacar la lealtad, el valor de la palabra, el esfuerzo permanente y el respeto por la comunidad.

2.2. Situación Actual

El centro de Formación Social BETHANIA pone a disposición de la comunidad su centro de servicios, en el que se llevan a cabo convenciones, cursos de formación, seminarios, talleres, convivencias, entre otros. Cada evento tiene una fecha de inicio y de fin, con las que se integra un calendario, manejado actualmente en registros físicos por los administradores.

Los trabajadores del centro son remunerados con un sueldo, sus tiempos de entrada y de salida son flexibles, y muchas veces trabajan horas extras pagadas, situación que usualmente causa confusión en los roles de pago.

Para optimizar tiempo y aumentar la calidad del servicio que ofrece el centro BETHANIA, se requiere de una solución tecnológica integral que permita llevar el registro de los eventos, los que ya se han llevado a cabo, los que se darán y la información que requiere cada uno para su planificación. Además, un calendario para la visualización de las fechas ocupadas y disponibles.

Es indispensable para la administración del centro que se maneje la entrada y salida de los trabajadores de forma digital, para evitar cualquier confusión que surja en el pago de sueldos incluidas horas extras, en el caso que existieran.

Se requiere además de un sitio web donde se exponga de forma gráfica las instalaciones y su disponibilidad, conjuntamente con los servicios de hospedaje y catering que se ofrecen.

2.3. Levantamiento de Procesos

Del análisis de la situación actual, se puede definir el siguiente mapa de procesos:

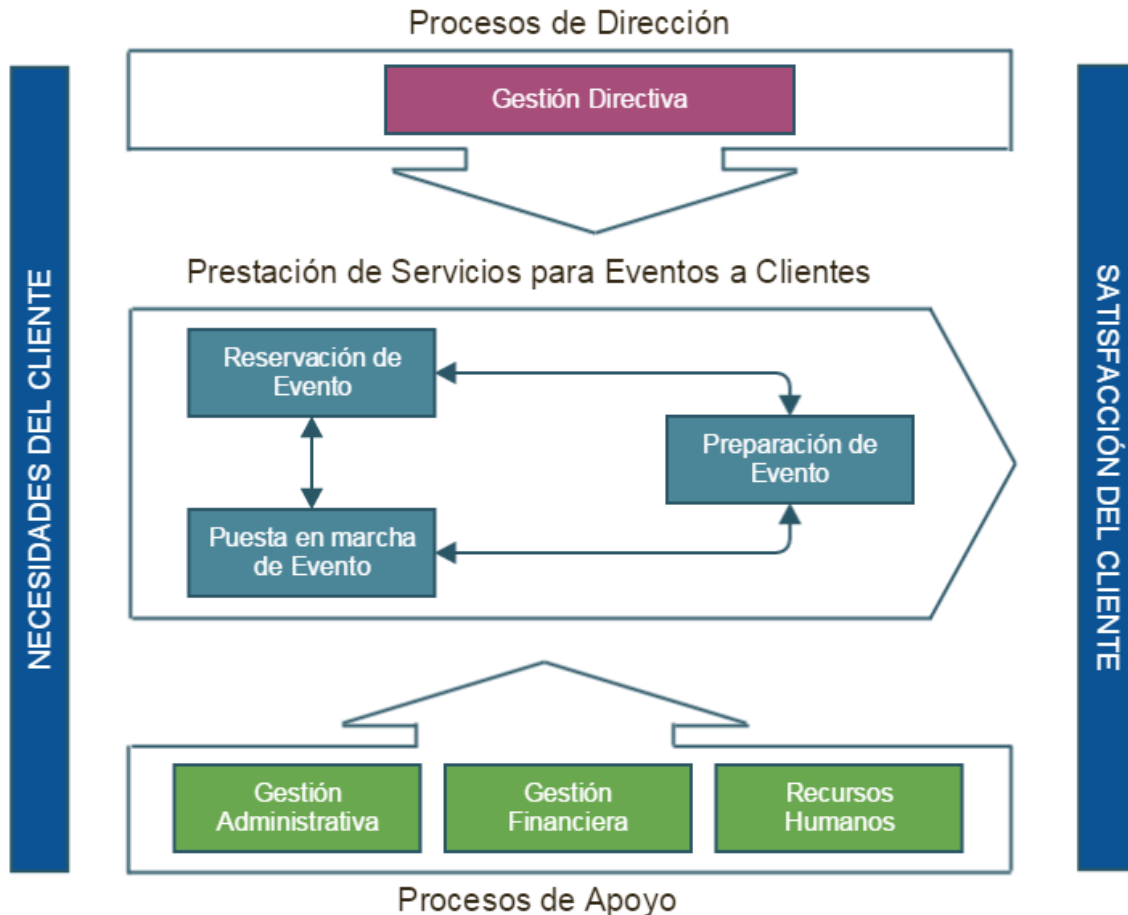


Figura 13 Diagrama de Procesos

Autor: Gustavo W Moyano

Cabe recalcar que éste diagrama ha sido tomado para representar al mayor nivel de abstracción posible.

En él se puede observar que la Prestación de Servicios para Eventos a Clientes es el proceso clave o principal, el que añade mayor valor al negocio.

Los procesos de Apoyo y Dirección permitirán el afinamiento de las entradas (necesidades del cliente) para que las salidas (satisfacción del cliente) sean las esperadas. En el presente caso de estudio, todos los subprocesos son

llevados a cabo por dos actores principales la administradora y los colaboradores del *Centro de Formación Bethania*.

2.3.1. Proceso Clave

El proceso clave consta de tres subprocesos principales:

2.3.1.1. Reservación de Eventos

Para éste subproceso se han considerado las actividades principales a realizarse, tal como lo indica la FIGURA 13:



FIGURA 14 Proceso de Reservación de Eventos

Autor: Gustavo W Moyano

Éste proceso es uno de los más importantes, uno de los que más entrega valor al negocio, por lo que será tomado en cuenta para su representación como diagrama de flujo:

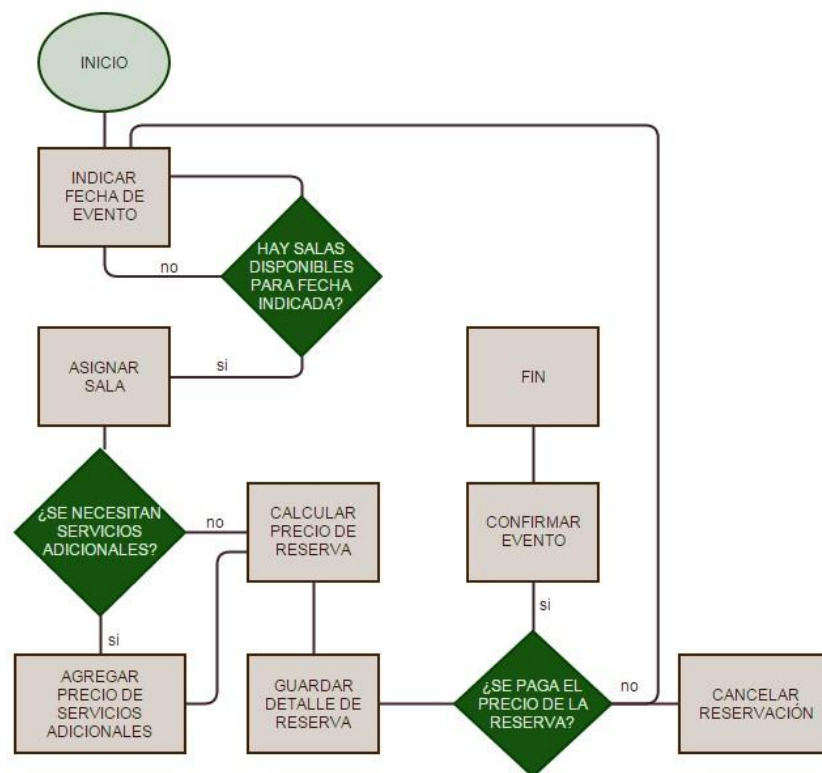


FIGURA 15 Diagrama de Flujo Reservación de Eventos

Autor: Gustavo W Moyano

En el desarrollo del presente proyecto, éste proceso será automatizado, e incluido en el sistema para su manejo de manera virtual.

2.3.1.2. Preparación de Evento

Éste subproceso permite llevar a cabo todas las actividades posteriores a la puesta en marcha del evento, se puede recalcar algunas actividades, tal como muestra la figura:



Figura 16 Proceso de Preparación de evento

Autor: Gustavo W Moyano

2.3.1.3. *Puesta en marcha de Evento*

En éste subproceso se encuentran las actividades que permiten la cobertura, apoyo y soporte que se brinda al cliente durante el evento, las actividades más destacadas son mostradas en la figura:



Figura 17 Proceso de Puesta en Marcha de Evento

Autor: Gustavo W Moyano

2.3.2. Procesos de Apoyo

Los procesos de apoyo en el presente caso ayudarán a que el proceso principal se lleve a cabo de una manera óptima y efectiva. Los procesos de apoyo que existen son:

2.3.2.1. *Gestión Administrativa*

Estará compuesto por el grupo de acciones que permiten mantener la administración del centro de formación en términos generales. En el caso de estudio, la gestión administrativa es ejecutada por una sola persona, la administradora general.

2.3.2.2. *Gestión Financiera*

En éste proceso están consideradas todas las acciones que velarán por la estabilidad y transparencia económica del centro de formación. La persona a cargo de éstas funciones es Maritza Benavides, la administradora de Bethania.

2.3.2.3. *Recursos Humanos*

Se necesita una administración de personal en todas las empresas. Bethania es considerada una organización también, por lo que el bienestar de los colaboradores debe ser puesto en primer lugar. La administradora de Recursos Humanos es la misma persona con roles administrativos y financieros en el presente caso de estudio.

2.3.3. Procesos de Dirección

El procesos de dirección que velará por la ejecución eficaz y eficiente del proceso principal es la Gestión Directiva, llevada a cabo en el presente caso por la Conferencia Episcopal Ecuatoriana.

3. CAPÍTULO 3 - SOLUCIÓN PROPUESTA / APLICACIÓN DAD

3.1. Ajuste de la Metodología

3.1.1. Definición de Metas

Basándose en la necesidad del cliente (en este caso el centro de formación social Bethania), se tendrán en cuenta las tres fases del ciclo de vida básico DAD: **Concepción**, **Construcción** y **Transición**, para la definición de las metas a ser cumplidas. En las tres fases se definirán aspectos dictados por la metodología DAD.

En la fase de **Concepción** se tienen los siguientes aspectos necesarios:

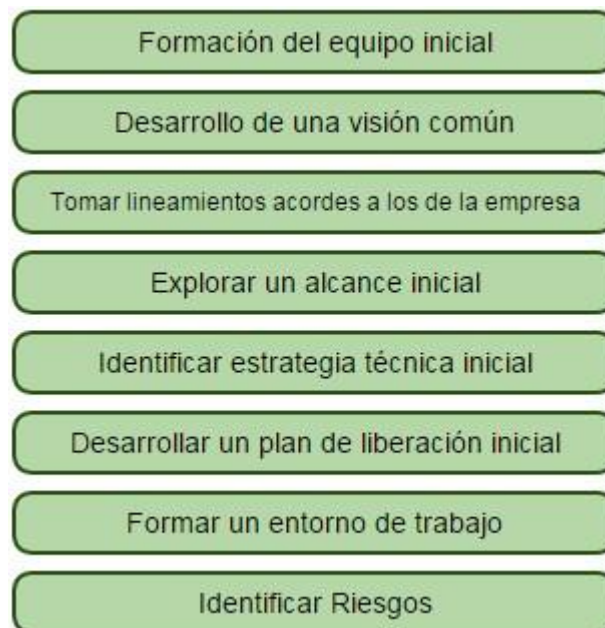


Figura 18 Metas de Concepción DAD

Autor: Scott W Ambler – Disciplined Agile Consortium

Para la **Construcción** se definirán:

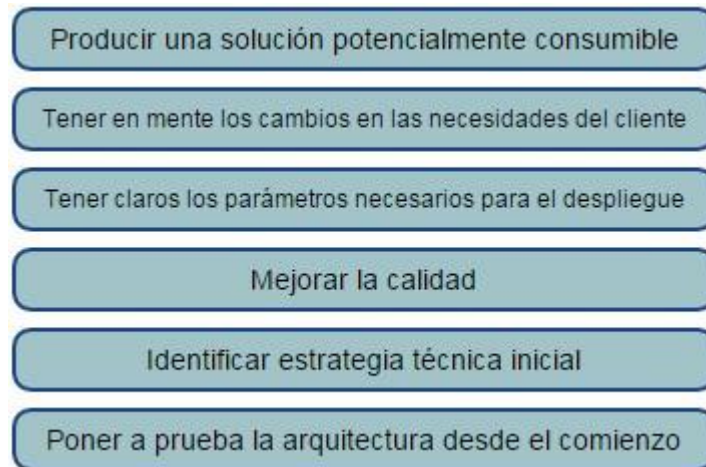


Figura 19 Metas de Construcción DAD

Autor: Scott W Ambler – Disciplined Agile Consortium

Y para la **Transición**:

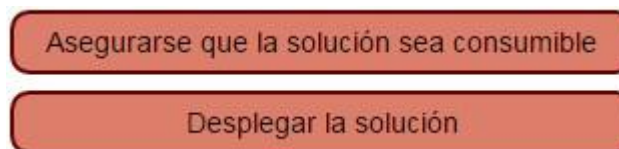


Figura 20 Metas de Transición DAD

Autor: Scott W Ambler – Disciplined Agile Consortium

3.1.2. Definición de ciclo de vida

El marco de trabajo Scrum es el recomendado para equipos ágiles en sus primeras experiencias, debido a que presenta su estructura con transparencia y simplicidad. Se han destacado los diferentes ciclos de vida que posee la metodología DAD, entre los cuales se encuentra el ciclo de vida ágil / básico, basado y extendido de Scrum. Éste ciclo permitirá utilizar las características estrellas de Scrum como: Sprint, Product Backlog, entre otras, traducidas en lenguaje DAD: Iteración, Lista de Ítems de Trabajo; conjuntamente con las características extras que ofrece DAD, como la cobertura de requerimientos no funcionales en la lista de ítems de trabajo y la definición clara de objetivos y metas a través de todo el desarrollo. Por tanto, se tomará el ciclo de desarrollo DAD ágil / básico que extiende a Scrum para llevar a cabo éste proyecto.

3.2. Iteración 0 – Concepción

En la primera iteración, se iniciará el proyecto y su definición mediante la fase de Construcción.

A continuación una breve descripción de las decisiones tomadas para ésta iteración, basándose en las metas de ésta fase:

- Formación del equipo inicial

No es necesario profundizar éste tema, debido a que el equipo se conforma por una persona. Cabe recalcar que se cumplirán diferentes roles en diferentes etapas del desarrollo.

- Desarrollo de una visión común

Éste aspecto es inalienable para dar un propósito al equipo, se responden preguntas como: ¿Qué estrategia se seguirá para desarrollar la visión? ¿Qué nivel de detalle se desea capturar? ¿A qué nivel de acuerdo se debe llegar con los clientes antes de pasar a la fase de Construcción? ¿Qué nivel de formalidad debe tener este acuerdo?

- Tomar lineamientos acordes a los de la empresa

Es necesario describir cómo el equipo se asegurará de que lo que está a punto de desarrollar se vea reflejado en la estrategia general de toda la organización. Se necesitará entender las estrategias técnicas y de negocio que son relevantes para la presente situación y se necesitará conocer qué bienes podrán contribuir a la causa. Se responden preguntas como: ¿Cuáles son los estándares y lineamientos que se deben seguir? ¿Qué plantillas se deben adoptar? ¿Cómo se tratará el tema de reúso de bienes tecnológicos ya existentes?

- Explorar un alcance inicial

Ésta meta del proceso de Concepción describe qué tan bien se capturan los requerimientos iniciales para la solución. Lo que se desea es hacer la cantidad de trabajo justa y necesaria para entender

lo que desea el cliente y los inversores, para empezar de una forma segura la fase de Construcción.

- Identificar estrategia técnica inicial

Esta meta describe que tan bien se identifican la estrategia o potenciales estrategias de arquitectura para producir la solución. Para ser efectivo, se necesitan considerar las siguientes preguntas: ¿A qué nivel de detalle se necesita llegar? ¿Qué modelos o vistas se deberían producir? ¿Qué enfoque de modelamiento se utilizará? ¿Cuál es la macro-estrategia para producir la solución?

- Desarrollar un plan de liberación inicial

Aunque los detalles aparezcan conforme se lleve a cabo la fase de Construcción, es necesario crear un plan inicial para el desarrollo. Es necesario tener consciencia del cómo se procederá al trabajo en grupo y del tiempo que se tomará.

- Formar un entorno de trabajo

Se necesitan responder las preguntas: ¿Cómo se organizará el espacio físico de trabajo? ¿Cómo se organizará el espacio virtual / electrónico de trabajo? ¿Qué herramientas se adoptarán para su manejo?

- Identificar Riesgos

Es necesario conocer cómo se inicializará el manejo de riesgos en el equipo, cómo serán estos riesgos documentados, qué tipo de riesgos se deben considerar y cuán detalladas deben ser sus descripciones.

3.2.1. Desarrollo de una visión común

Para esta meta se tiene lo siguiente:

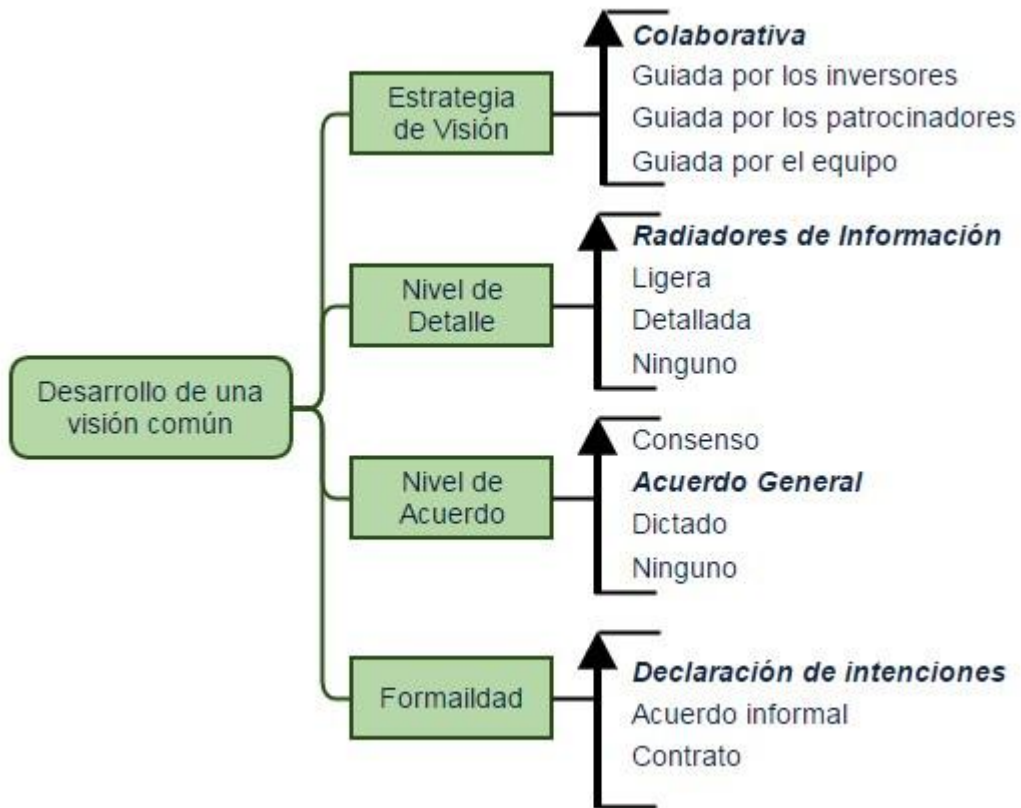


Figura 21 Meta Desarrollo de una visión común

Autor: Scott W Ambler – Disciplined Agile Consortium

La estrategia de visión escogida para éste caso es la Colaborativa, mediante la cual se logra definir un objetivo en común conjuntamente con el cliente.

Se necesita un nivel de detalle alto, en el que se cubran varios niveles de abstracción para la resolución de los problemas y la automatización de los procesos.

Se tendrán acuerdos generales con el cliente, en el que se informarán de manera clara y concisa los objetivos y las actividades a llevar a cabo. La formalidad de estos acuerdos se verá reflejada en una declaración de intenciones, ya sea por escrito o de forma oral.

Después de un análisis de la situación actual conjuntamente con la administradora del centro de formación Bethania se llegó a un acuerdo de beneficio bilateral, en el que, el desarrollador se compromete a entregar una solución consumible que permita cubrir las necesidades provistas, y por otro lado, la administradora de Bethania se compromete a proveer de la información necesaria para tener un entendimiento a varias capas de abstracción del funcionamiento y la dinámica de la organización.

3.2.2. Explorar un alcance inicial

Para esta meta se tiene:

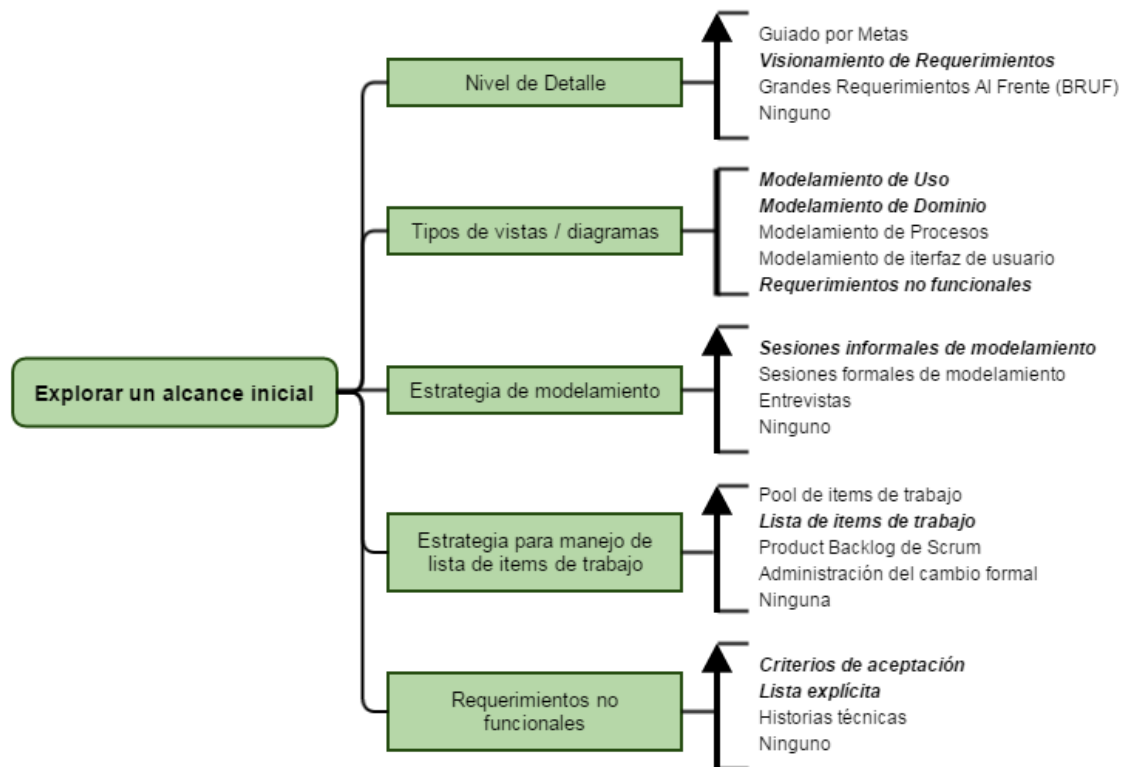


Figura 22 Meta Explorar un alcance inicial

Autor: Scott W Ambler – Disciplined Agile Consortium

El nivel de detalle escogido será el Visionamiento de Requerimientos.

3.2.2.1. *Visionamiento de Requerimientos*

Para la definición de este punto en el presente caso de uso, se ha tomado como base la situación actual de la empresa, que tiene tres frentes principales:

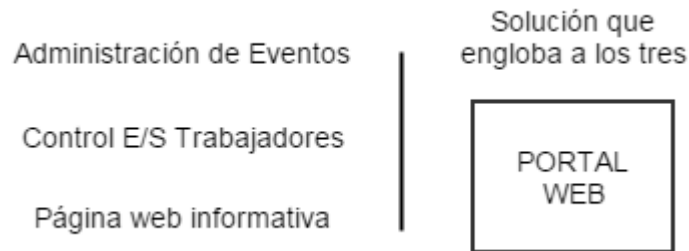


Figura 23 Frentes de trabajo

Autor: Scott W Ambler – Disciplined Agile Consortium

Un portal web es la herramienta que permitirá navegar entre páginas web, y permitirá atacar los tres frentes principales de trabajo de una manera centralizada. Además de ser la solución idónea técnicamente, cumple las expectativas de Bethania.

Para profundizar el conocimiento acerca del sistema que se creará, se necesitará **modelamiento de uso**, **modelamiento de dominio** y posiblemente un modelamiento inicial de las **interfaces de usuario** como Tipos de Vistas / Diagramas.

3.2.2.2. *Modelamiento de Uso*

Como dicho anteriormente, se tiene tres frentes diferentes de trabajo, serán cubiertos uno a uno. Las funcionalidades en general cuyo uso necesita ser modelado son:

Para el frente de administración de eventos se considerará:

- Administración de Eventos (CU1)
- Administración de Clientes (CU2)

Para el frente de control de entrada y salida de trabajadores se tendrán:

- Administración de trabajadores (HU1)

- Control de tiempos de entrada y salida (HU2)

Existen diferentes formas de realizar modelamiento de uso, entre las cuales están: casos de uso (CU) e historias de usuario (HU)²¹.

A lo largo del modelamiento inicial, serán utilizadas las dos técnicas diferentes.

3.2.2.2.1. Caso de Uso General

El caso de uso general será tal y como se muestra en la figura:

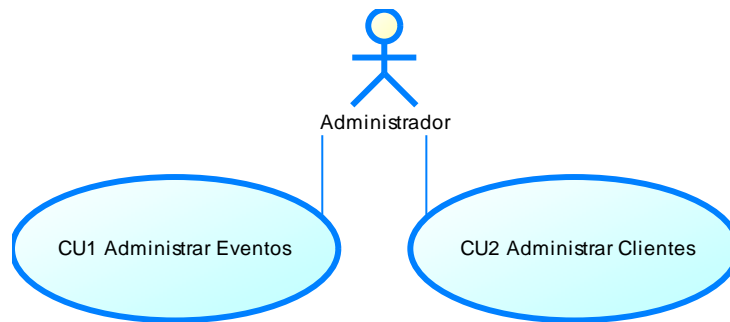


Figura 24 Caso de Uso General

Autor: Gustavo W Moyano

En el caso de uso general se tendrá en cuenta un solo actor: el administrador del sistema, para el caso de estudio, Maritza Arias la administradora de Bethania.

3.2.2.2.2. Administración de Eventos (CU1)

En la figura se puede ver los casos de uso en los que se descompone la funcionalidad de Administrar Eventos:

²¹ Técnica de modelamiento de uso en metodologías ágiles de desarrollo, es un requerimiento de software representado en lenguaje común de usuario. Se enfoca en definir 'quién', 'qué' y 'por qué'.

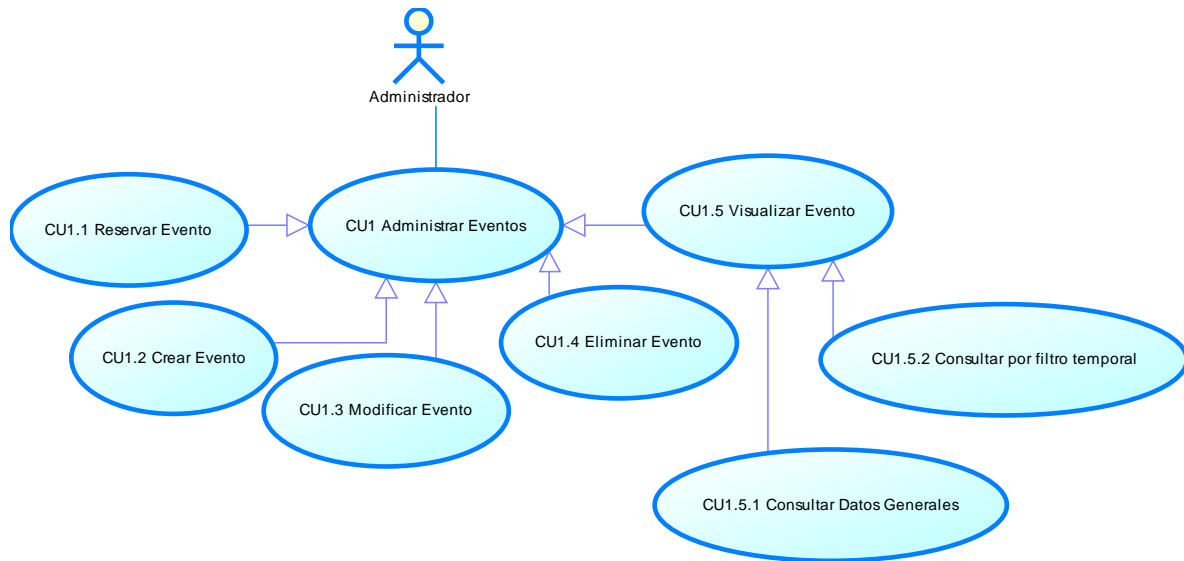


Figura 25 Caso de Uso 1 - Administración Eventos

Autor: Gustavo W Moyano

El administrador será el que lleve a cabo todas las funcionalidades que heredan de Administrar Eventos:

- **CU1.1 Reservar Evento**

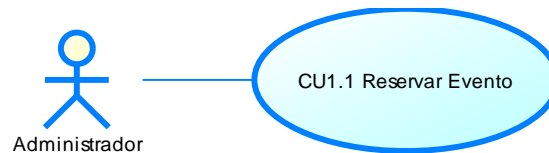


Figura 26 CU1 - Siguiendo Nivel 1 - Reservar Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador presiona el botón "Reservar Evento".
- 1 El sistema desplegará una ventana para el ingreso de los datos de evento y de cliente.
- 2 El administrador escogerá uno de entre los clientes mostrados.
- 3 El administrador ingresará todos los datos de evento.
- 4 El sistema realizará validaciones pertinentes (E1)

- 5 El administrador pulsa el botón “Reservar”.
- 6 El sistema crea un evento y cierra ventana de Reservación.

➤ Excepciones

- E1: No cumple con la Regla de validación.

➤ Flujo Alternativo

- 1 Dirigirse a otra ventana, de ser el caso que no se requiera reservar.
- 2 Crear Cliente en el caso que no exista el deseado.
- 5 El administrador pulsa “Salir”, se muestra confirmación antes de salir.

- **CU1.2 Crear Evento**

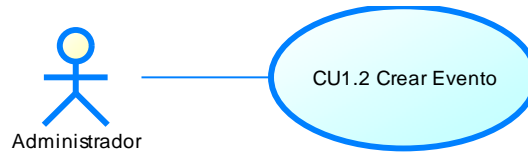


Figura 27 CU1 - Siguiendo Nivel 2 - Crear Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador presiona el botón “Crear Evento”.
- 1 El sistema despliega una ventana para el ingreso de los datos de evento.
- 2 El administrador ingresa todos los datos de evento.
- 3 El sistema realiza validaciones pertinentes (E1)
- 4 El administrador pulsa el botón “Crear”. (E2)
- 5 El sistema crea un evento y cierra ventana de Creación.

➤ Excepciones

- E1: No cumple con la Regla de validación.
- E2: Error al crear.

➤ Flujo Alternativo

- 1 Dirigirse a otra ventana, de ser el caso que no se requiera crear.
- 4 El administrador pulsa “Salir”, se muestra confirmación antes de salir.

- **CU1.3 Modificar Evento**

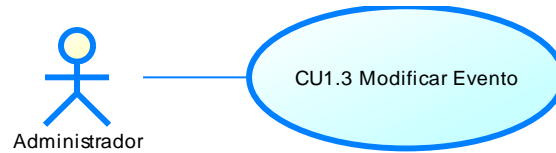


Figura 28 CU1 - Siguiendo Nivel 3 - Modificar Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador da 'click' sobre un evento ya creado/reservado.
- 1 El sistema despliega una ventana modal para la modificación de los datos de evento.
- 2 El administrador ingresa los datos del evento a ser modificado.
- 3 El sistema realiza validaciones pertinentes (E1)
- 4 El administrador pulsa el botón "Guardar". (E2)
- 5 El sistema guarda modificaciones y cierra ventana de "Modificación".

➤ Excepciones

- E1: No cumple con la Regla de validación.
- E2: Error al Guardar

➤ Flujo Alternativo

- 1 Dirigirse a otra ventana, de ser el caso que no se requiera modificar.
- 4 El administrador pulsa "Salir", se muestra confirmación antes de salir.

- **CU1.4 Eliminar Evento**

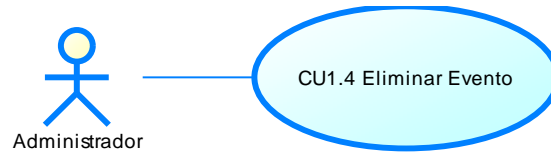


Figura 29 CU1 - Siguiendo Nivel 4 - Eliminar Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador da 'click' sobre un evento ya creado/reservado.
- 1 El sistema desplegará una ventana modal para la modificación de los datos del evento.
- 2 El administrador pulsará el botón "Eliminar". (E1)
- 3 El sistema pide confirmación de acción.
- 4 El administrador confirma la eliminación.
- 5 El sistema elimina el evento y cierra ventana de Modificación/Eliminación.

➤ Excepciones

- E1: Error al eliminar.

➤ Flujo Alternativo

- 1 Dirigirse a otra ventana, de ser el caso que no se requiera eliminar.

- **CU1.5.1 Consultar Datos Generales**

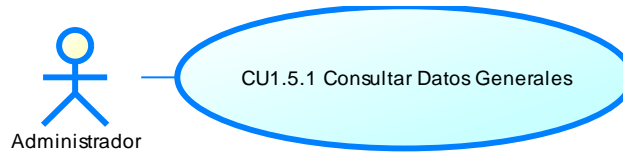


Figura 30 CU1 - Siguiendo Nivel 5 - Consulta Datos Generales Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador da click en “Admin”
- 1 El sistema despliega de manera visual todos los eventos en el calendario

- **CU1.5.2 Consultar por filtro temporal**

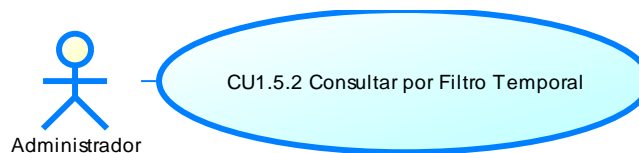


Figura 31 CU1 - Siguiendo Nivel 6 - Consulta filtro tiempo Evento

Autor: Gustavo W Moyano

➤ Flujo Principal

- 0 El administrador da ‘click’ en “Admin”
- 1 El sistema despliega de manera visual los eventos en el calendario.
- 2 El actor da ‘click’ en Año, Mes, Semana o Día.
- 3 El sistema despliega los eventos que se llevarán a cabo, dependiendo de la selección.

3.2.2.2.3. Administración de Clientes (CU2)

En la figura se pueden observar los casos de uso en los que se descompone la funcionalidad de Administrar Clientes:

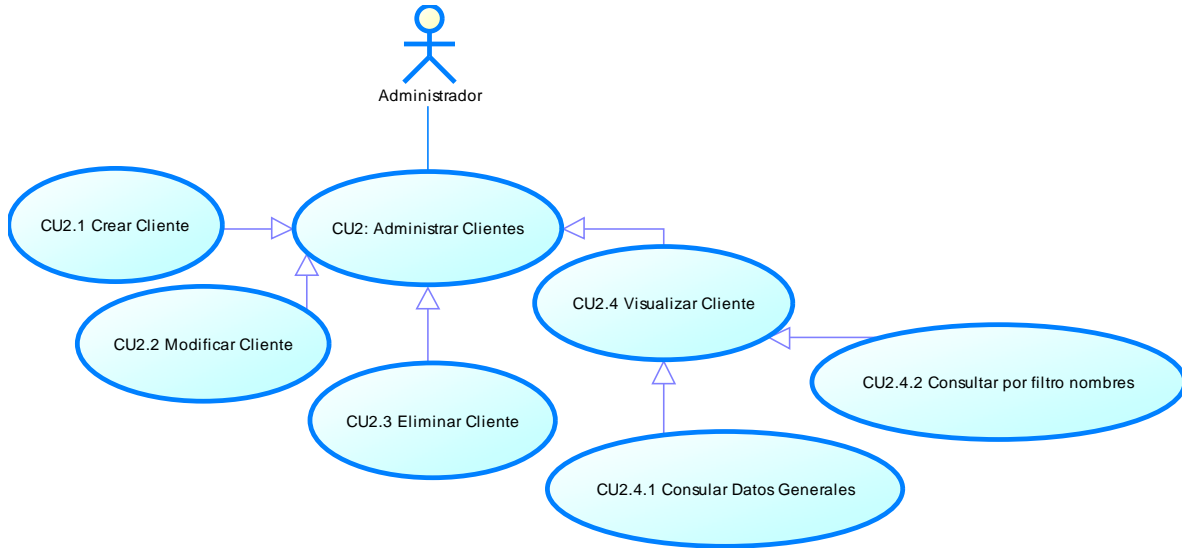


Figura 32 Caso de Uso 2 - Administración de Clientes

Autor: Gustavo W Moyano

El administrador será el que lleve a cabo todas las funcionalidades que heredan de Administrar Clientes:

- CU2.1 Crear Cliente
- CU2.2 Modificar Cliente
- CU2.3 Eliminar Cliente
- CU2.4 Visualizar Cliente
- CU2.4.1 Consultar Datos Generales
- CU2.4.2 Consultar por Filtro Nombres

El comportamiento del sistema será similar al de la administración de eventos, con sus diferencias en las consultas por filtro de nombre que se realizarán para los clientes.

3.2.2.2.4. Administración de Trabajadores (HU1)

Para mejor entendimiento de la funcionalidad de Administración de trabajadores, se definirán los siguientes perfiles o 'personas'²²:


Nombre	Detalles	Objetivo
 Maritza Arias - Administradora Bethania *Primaria*	<p>Maritza, una mujer dedicada a su trabajo al 100%, dispuesta a colaborar con su comunidad cuando la necesiten.</p> <p>Maritza cubre diversos roles en la organización, teniendo a cargo la parte administrativa, dividida en varios frentes: gerente de capital humano, de finanzas, de planificación y logística de eventos y contacto con el cliente.</p>	<p>Maritza desea optimizar el tiempo ocupado en sus tareas.</p> <p>Desea reemplazar su carpeta de manejo de eventos por una solución automatizada que se pueda manejar desde su ordenador.</p> <p>Desea optimizar la tarea de cálculo de horas de trabajo extras de sus colaboradores.</p>

Figura 33 Definición de Personas – Administradora

Autor: Gustavo W Moyano


Nombre	Detalles	Objetivo
 Colaborador Bethania	<p>Un colaborador puede cumplir varios roles funcionales en la organización: Apoyo en logística de eventos, catering, preparación de ambientes (salas y habitaciones), adquisición de productos.</p> <p>El colaborador trabaja una jornada laboral de 8 horas, sin embargo existen ocasiones en las que se extiende su tiempo de trabajo.</p> <p>El colaborador recibe un pago por horas de trabajo</p>	<p>El colaborador desea registrar sus horas de trabajo.</p> <p>El colaborador desea que sus horas de trabajo extras sean compensadas.</p>

Figura 34 Definición de Personas – Colaborador

²² Técnica para capturar conocimiento acerca de un usuario o un cliente del producto.

Autor: Gustavo W Moyano

Teniendo una definición previa de las personas, definimos efectivamente la historia de usuario:

Título:	Administración de trabajadores	HU1
Como	administradora de Bethania	
Deseo	mantener un registro de trabajadores	
Para	organizar su información de manera efectiva	

Figura 35 Historia de Usuario 1 - Administración trabajadores

Autor: Gustavo W Moyano

Ésta historia de usuario (HU), refleja la funcionalidad en su mayor nivel de abstracción, es decir a su nivel macro. En un siguiente paso esta HU será descompuesta en HUs claras, factibles y probables.

3.2.2.2.5. Control de tiempos de entrada y salida (HU2)

Para el entendimiento más profundo de ésta HU, se utilizará las personas previamente definidas en la HU1, teniendo la siguiente historia de usuario:

Título:	Control de tiempos de entrada y salida	HU2
Como	administradora de Bethania	
Deseo	controlar el tiempo de entrada y salida de los trabajadores	
Para	facilitar el cálculo de sus horas extras	

Figura 36 Historia de Usuario 2 - Control tiempos e/s

Autor: Gustavo W Moyano

Definida de igual manera a nivel macro, posteriormente dividida en HUs más pequeñas.

3.2.2.3. Modelamiento de Dominio

3.2.2.3.1. Administración de Eventos

Para el caso de la funcionalidad principal (englobada en los dos primeros casos de uso CU1 y CU2), se tienen las siguientes entidades:

Cliente: contiene los datos del cliente que necesita llevar a cabo un evento y realizar el pedido al centro Bethania para la reservación de su espacio en una fecha definida.

Evento: representa al evento como tal, tiene su fecha de inicio y de final, el cliente que llevará a cabo el evento y otros datos informativos.

Espacio: representa la sala o la habitación que serán utilizadas por los participantes del evento.

Grupo: existen grupos a los que pertenecen las habitaciones, son netamente una guía que permitirá acercar al sistema mucho más a la realidad.

Después de la definición de las entidades, se necesita saber cómo se relacionan. Su relación puede visualizarse en el siguiente diagrama:

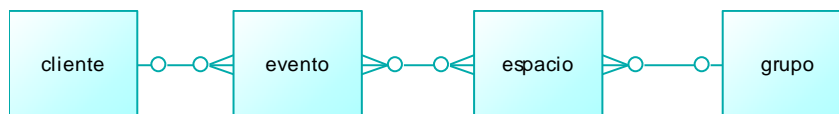


Figura 37 Admin Eventos - Relación de Entidades

Autor: Gustavo W Moyano

La

manera concreta de entender éste diagrama es apegándola a la situación actual. Se tendrá un evento con un cliente involucrado, y un espacio (sea una sala o una habitación), y un espacio puede pertenecer a un grupo.

El siguiente paso será añadir atributos correspondientes al diagrama de entidades, se utilizará directamente un diagrama conceptual de base de datos:

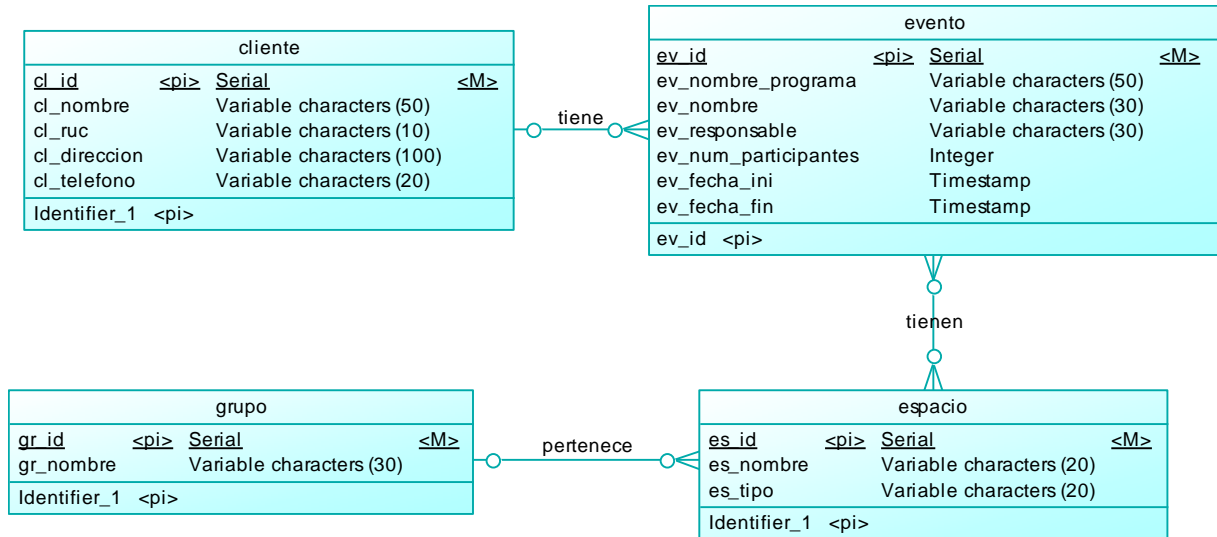


Figura 38 Admin Eventos - Diagrama Conceptual de Base de Datos

Autor: Gustavo W Moyano

Aquí se ve reflejada información relevante que necesitará cada entidad. Además de éste diagrama se puede extraer el siguiente diagrama físico de base de datos:

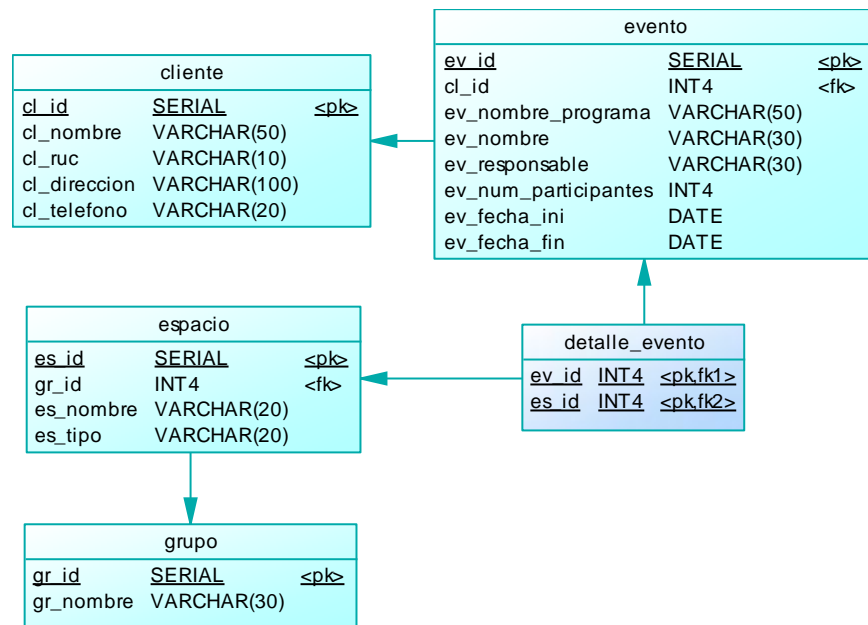


Figura 39 Admin Eventos - Diagrama Físico de Base de Datos

Autor: Gustavo W Moyano

Para la base de datos, se tiene el manejo de las entidades y sus relaciones basadas enteramente en el mapa físico de base de datos.

3.2.2.3.2. Control Entrada/Salida de Trabajadores

Para la funcionalidad de control E/S de trabajadores, se tiene el siguiente enfoque:

Entidades:

Trabajador: Ésta entidad contendrá la información de un trabajador, el rol que desempeña y su horario planificado de trabajo (usualmente 8 horas diarias).

Día de trabajo: Contiene un identificador del trabajador, la fecha del día, la hora de entrada (login) y la hora de salida (logout) actuales.

La forma en la que se relacionan será de la siguiente manera:



Figura 40 Control E/S - Relación de Entidades

Autor: Gustavo W Moyano

Después para el diagrama conceptual de base de datos:

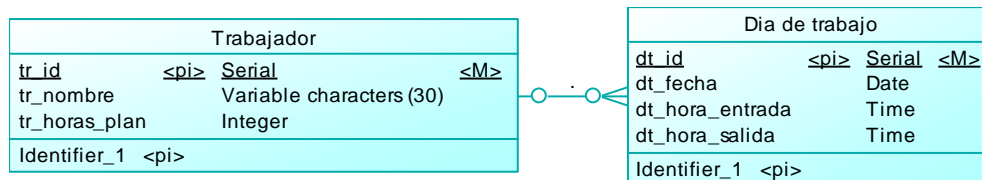


Figura 41 Control E/S Diagrama Conceptual de Base de Datos

Autor: Gustavo W Moyano

Que genera el siguiente diagrama físico:

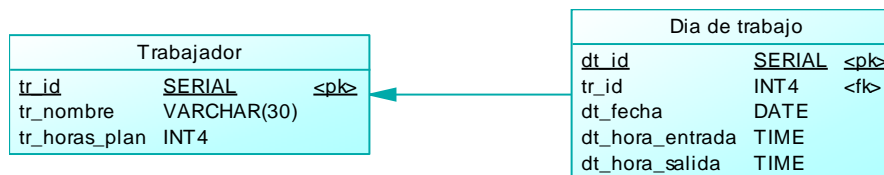


Figura 42 Control E/S Diagrama Físico de Base de Datos

Autor: Gustavo W Moyano

3.2.2.4. Sesiones informales de modelamiento

El modelamiento que se ha llevado a cabo fue construido en diversas sesiones de modelamiento junto al cliente. El objetivo ha sido siempre mantener al sistema lo más cercano a la realidad y automatizar los procesos de tal manera que se optimice la entrega de valor del negocio.

3.2.2.5. Lista de ítems de trabajo

La estrategia que se utilizará para generar la lista de ítems de trabajo será la siguiente:

- Se manejarán principalmente dos tipos de ítems de trabajo:
 - Requerimientos Funcionales
 - Requerimientos No Funcionales

Explicados en la siguiente Figura:

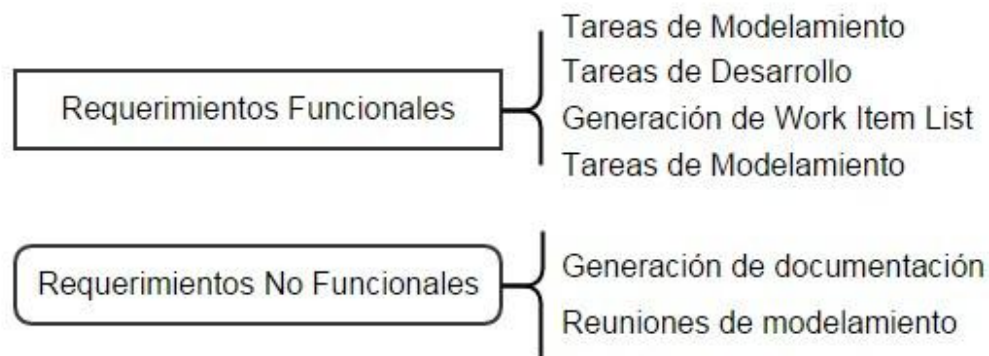


Figura 43 Tipos de Ítems de Trabajo

Autor: Gustavo W Moyano

En la figura se han expuesto algunos ejemplos de lo que se podrá encontrar en la lista de ítems de trabajo, es posible que en la etapa de construcción se creen nuevos tipos de tareas.

El funcionamiento de la lista de ítems de trabajo será de ordenamiento por prioridad:

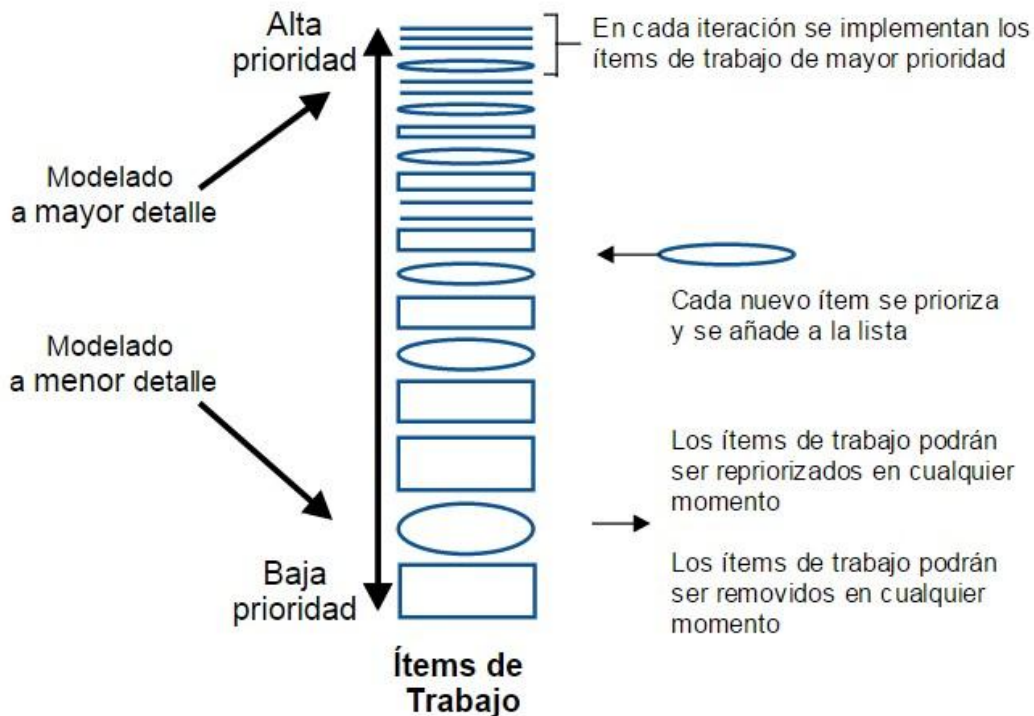


Figura 44 Funcionamiento de Lista de Ítems de Trabajo

Autor: Gustavo W Moyano

Debido a que la lista de ítems de trabajo contendrá requerimientos funcionales y no funcionales, serán ordenados por prioridad y pasarán a la lista de ítems de trabajo de iteración.

Como es explicado en la figura, la lista tendrá su dinámica, en la que cada nuevo ítem de trabajo podrá ser 'repriorizado' o removido en cualquier momento.

3.2.2.6. *Requerimientos no funcionales (Lista explícita)*

Dentro de la lista de requerimientos no funcionales sin duda estarán todos los criterios que necesitan ser definidos y que no forman parte de la ejecución del sistema. Aunque su nombre sea un tanto erróneamente interpretado, los requerimientos no funcionales son vitales para completar el ciclo de vida del sistema. En el presente caso, la lista de requerimientos no funcionales será:

- Estructura de documentos
- Objetivos del sistema
- Acuerdos de modelamiento de sistema

- Acuerdos de funcionamiento de sistema
- Usabilidad del sistema
- Escalabilidad del sistema

Cada frente será atacado a partir de las acciones que definimos en la lista de ítems de trabajo (Generación de Documentación y Reuniones de modelamiento en general).

3.2.3. Identificar estrategia técnica inicial

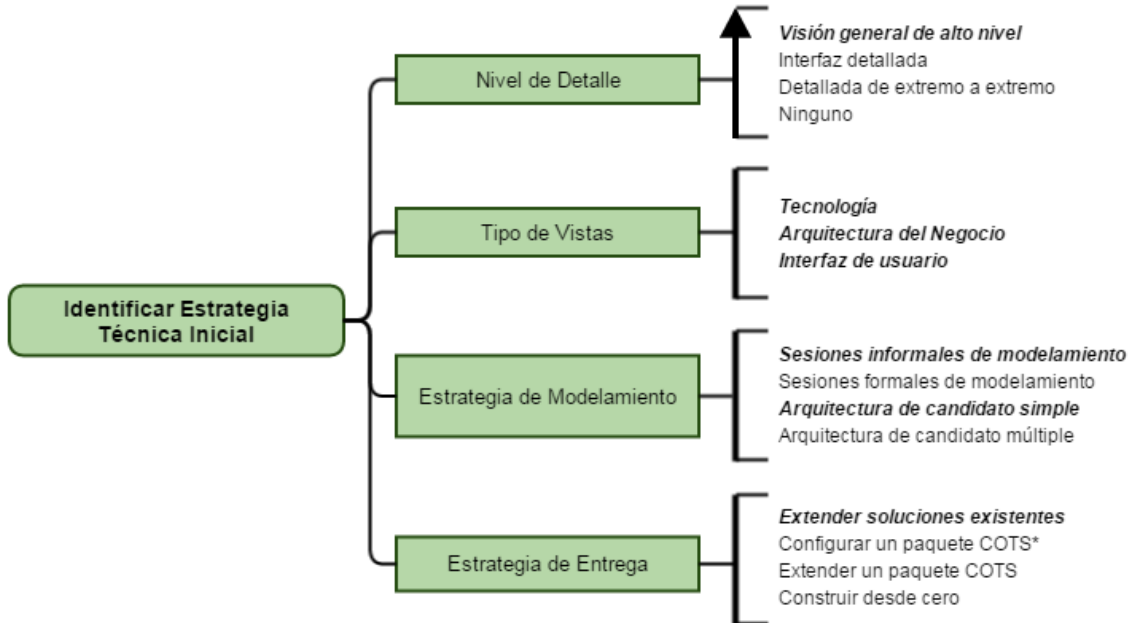


Figura 45 Meta Identificar Estrategia Técnica Inicial

Autor: Scott W Ambler – Disciplined Agile Consortium

El nivel de detalle utilizado será una Visión general de alto nivel, debido a que se necesita conceptualizar de manera simple y macro las funcionalidades.

En cuanto al tipo de vistas:

3.2.3.1. Arquitectura del Negocio

Desde un enfoque técnico, la aplicación web utilizará la siguiente arquitectura:

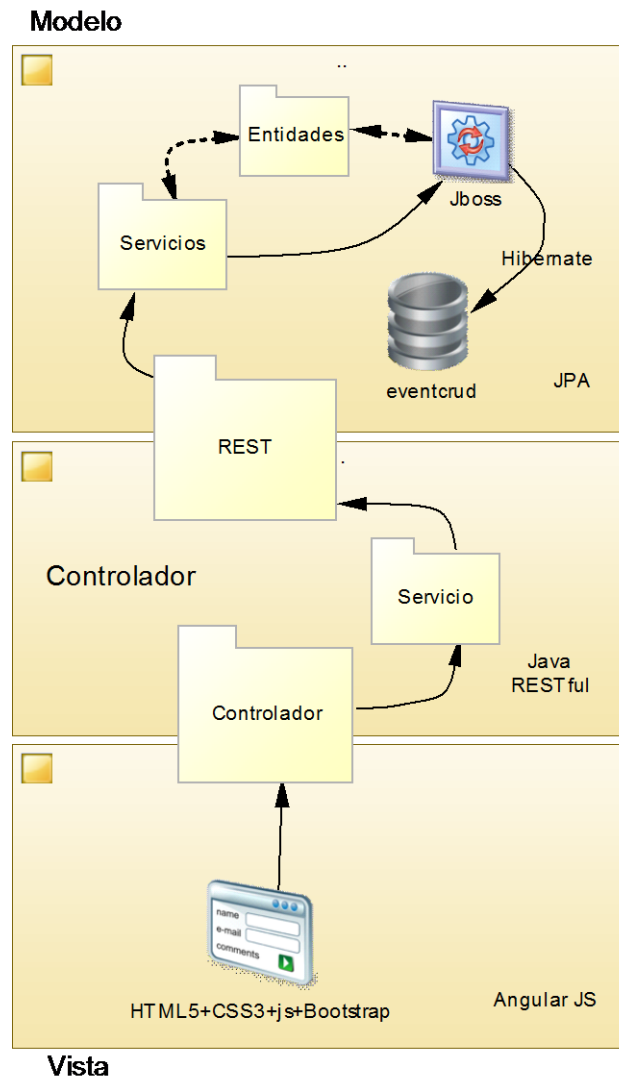


Figura 46 Arquitectura de Aplicación

Autor: Gustavo W Moyano

A nivel macro, existen tres áreas de arquitectura:

- Modelo
- Controlador
- Vista

3.2.3.2. Interfaces de Usuario

Visualmente, las interfaces de usuario o prototipos permiten tener un acercamiento a lo que será en última instancia del sistema. Para éste caso, se iniciará con la concepción del mapa del sitio web:

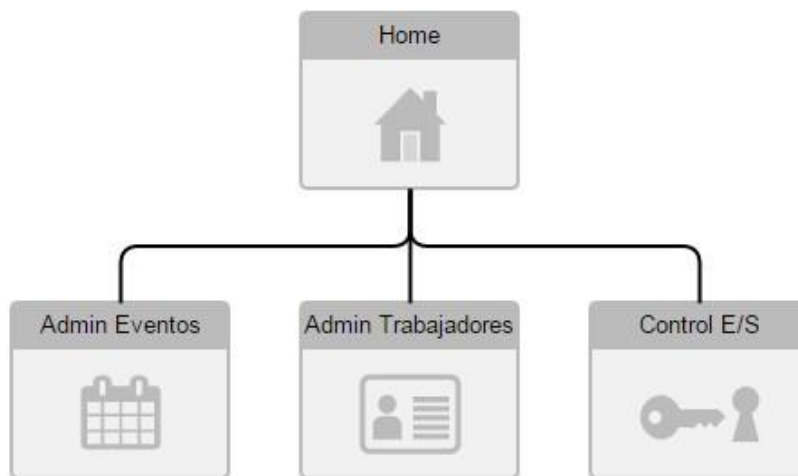


Figura 47 Mapa web de aplicación

Autor: Gustavo W Moyano

La página principal (Home) tendrá información acerca del Centro de Formación Bethania, conjuntamente con links a las otras páginas en forma de una barra de navegación.

La página de Administración de Eventos, permitirá apreciar de manera visual la planificación en lo concerniente a fechas para los eventos:

Interfaz de usuario para la administración de eventos. La barra de navegación superior contiene los enlaces: Home, Admin Eventos (seleccionado) y Control E/S Trabajadores.

Calendario de Eventos

El calendario muestra el mes de ABRIL. Las fechas están representadas en un grid de 5 columnas y 5 filas. Las fechas 15, 16, 17 y 18 están resaltadas en rojo, indicando eventos reservados. Las fechas 13, 14, 17 y 18 también están resaltadas en azul.

Formulario de Reservación Eventos:

- Cliente:
- Sala:
- Catering: ☒ Habitación: ☒
- Fecha inicio:
- Fecha fin:
- Botón: Reservar

Figura 48 Prototipo de Calendario de Eventos

Autor: Gustavo W Moyano

3.2.3.3. *Arquitectura de candidato simple*

Para la concepción del sistema, se ha tenido en mente la arquitectura de candidato simple. La característica principal de este tipo de arquitectura es su simplicidad y la facilidad con la que puede ser comprendida. En una arquitectura de candidato simple.

3.2.4. Desarrollar un plan de liberación inicial

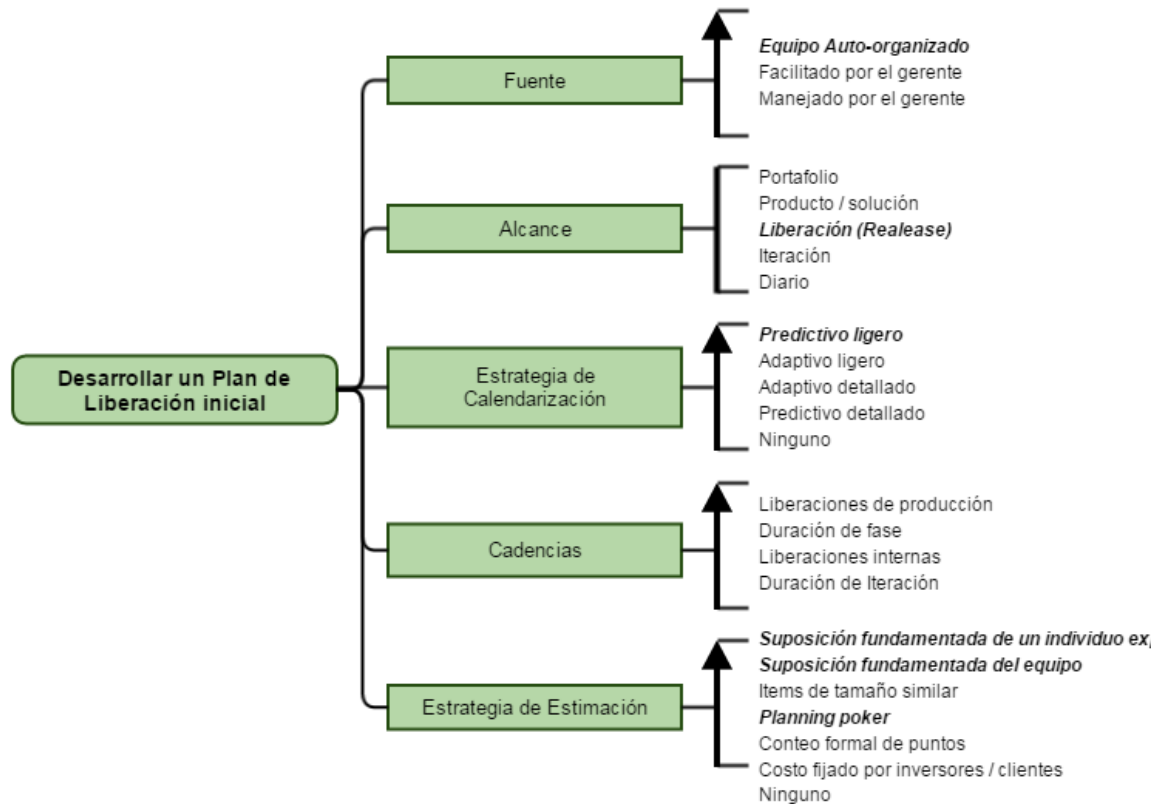


Figura 49 Meta Desarrollar un Plan de Liberación Inicial

Autor: Scott W Ambler – Disciplined Agile Consortium

3.2.4.1. *Equipo Auto-Organizado*

En el caso tratado, el equipo consta de una sola persona que deberá asumir todos los roles que conforman a un equipo de desarrollo ágil, en el que la auto-organización es una de las características claves para lograr cumplir con todos los objetivos de manera eficiente.

3.2.4.2. *Alcance*

El alcance inicial del sistema será una Liberación (Release) parcial en la que se observará las funcionalidades nucleares que dan más valor al negocio.

3.2.4.3. *Estrategia de Calendarización*

La duración de la puesta en marcha de los ítems de trabajo dentro de la lista será definida mediante una predicción ligera, que permita visualizar a un alto nivel de abstracción el total del tiempo a ser utilizado para la culminación del sistema.

4. CAPÍTULO 4 – CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

- La metodología de desarrollo ágil ‘Disciplined Agile Delivery’ (DAD) permite desarrollar aplicaciones de software siguiendo lineamientos de ‘agilismo’ apropiados también para escalar desarrollo ágil a grandes empresas.
- DAD ofrece varios ciclos de vida de software a aplicarse, con los que se abre la puerta a la personalización de las técnicas de desarrollo para una empresa casi en su totalidad.
- El desarrollo por cumplimiento de metas diseñado por DAD obliga al desarrollador a realzar una de las características más buscadas entre los interesados en el desarrollo de software: la disciplina.
- Dentro de las prácticas DAD se encuentra una de las más importantes en cuanto a la aplicación en grandes organizaciones, la conciencia empresarial. Esto permite a los agilistas dentro de DAD tener múltiples niveles de conciencia al momento de desarrollar, realzando así la eficiencia y efectividad de las soluciones ofrecidas.
- El manifiesto ágil disciplinado expone el núcleo de la mentalidad de desarrollo ágil, permitiendo un completo entendimiento de los principios y valores que rodean a ésta metodología.
- La ventaja de utilizar DAD por sobre otro marco de trabajo es notoria el momento de enfocar de manera técnica el desarrollo del sistema, aunque para DAD sea muy complicado encontrar nichos de mercado apropiados.
- Los ciclos de vida de software que ofrece DAD conforman una gama de diferentes opciones, dejando a criterio de los equipos ágiles de desarrollo DAD la elección del que se apegue más a las necesidades del proyecto.
- El “*work item pool*” de DAD abarca requerimientos no funcionales, lo que representa una ventaja considerable sobre el manejo por “*product backlog*” que ofrece *Scrum*.
- Las herramientas: Java Persistence API, Java Restful Services y AngularJS, ofrecen una arquitectura sólida para el desarrollo web.

- Las historias de usuario y los casos de uso nos permiten apreciar visualmente las potenciales funcionalidades del sistema.
- El desarrollo mediante la utilización de definición de historias de usuario, tareas y sub-tareas, facilita en gran manera el trabajo al desarrollador, debido a que se construye un entendimiento previo a la programación.
- En próximos *releases* de la aplicación se implementarán funcionalidades adicionales como exportar reportes como archivos externos, administración de costos y precios de eventos, administración de espacios, entre otras.
- La automatización de procesos para las organizaciones y empresas de nuestros tiempos es inevitable, porque permite ahorrar lo más valioso: el tiempo.
- El centro de Formación BETHANIA tiene bajo su mano una serie de tareas que pueden ser automatizadas y en un futuro pueden ahorrar mucho tiempo, potencialmente útil.
- La agenda de la administradora de BETHANIA está 100% cubierta a través de todos los días laborables. La labor cumplida es exhaustiva y constante para lograr cubrir todos los roles y responsabilidades.

4.2. Recomendaciones

- Utilizar el acercamiento ofrecido por DAD para la implementación de marcos de trabajo de desarrollo de software ágiles en grandes empresas.
- Escoger de manera detenida uno de los ciclos de vida de software ofrecidos por DAD que más se apegue a las necesidades del proyecto.
- Profundizar el conocimiento acerca de procesos del negocio para tener buenas bases previas a cualquier especificación de requerimientos, diseño o de software.
- Una vez tomada la decisión de utilizar herramientas ágiles de desarrollo, tener siempre en mente el manifiesto ágil y los valores del “agilismo”.
- Adoptar la filosofía de desarrollo ágil durante los estudios de ingeniería de software. Crear una materia en el pensum que cubra los conocimientos antes mencionados.
- Utilizar la metodología DAD para adoptar metodologías ágiles a gran escala en grandes empresas.
- Utilizar la técnica de un *work item list* antes que la de *product backlog* para tener un mayor manejo sobre el desarrollo y la puesta en marcha del proyecto.
- Si se necesita implementar un proyecto basado en tecnología web, utilizar las herramientas de Java: Persistence Api y Restful services en conjunto con tecnologías Javascript: Angular y HTML5+CSS3.
- Administrar el tiempo de manera efectiva si se maneja un entorno de alta presión y stress, para lograr cubrir todas las actividades y responsabilidades de múltiples roles en una empresa.
- En un ambiente con múltiples roles, automatizar tareas con el objetivo de optimizar la utilización del mismo.
- El manejo de la aplicación es totalmente intuitivo y su uso no implica ningún tipo de conocimiento técnico previo.

BIBLIOGRAFÍA

- [1 AMBLER, S. & LINES, M.. (2011). Disciplined Agile Delivery - An agile process
] decision framework for the enterprise. noviembre 25, 2014, de DISCIPLINED AGILE
CONSORTIUM Sitio web: <https://disciplinedagiledelivery.wordpress.com/>

- [2 CUNNINGHAM, W.. (2001). Manifesto for Agile Software Development. noviembre 28,
] 2014, de Main Agilists Sitio web: <http://agilemanifesto.org/>

- [3 Oracle. (2005). JAX-RS 2.0 API Specification (Rev a). enero 3, 2015, de Oracle and/or
] its affiliates Sitio web: <https://jax-rs-spec.java.net/nonav/2.0-rev-a/apidocs/index.html>

- [4 WIUM, H.. (2011). CSS Tutorial. enero 4, 2015, de Refsnes Data Sitio web:
] http://www.w3schools.com/css/css3_intro.asp

- [5 Jendrock, E., Ball J., Carson D., Evans I., Fordin S. & Haase K.. (2010). The Java EE 5
] Tutorial. enero 16, 2015, de Oracle and/or its affiliates Sitio web:
<http://docs.oracle.com/javaee/5/tutorial/doc/?wp406183&PersistenceEJB.html#wp78460>

ANEXOS

Abreviaturas	Definiciones
DAD	Disciplined Agile Delivery (Entrega Ágil y Disciplinada)
Java SE	Java™ Standard Edition
Java EE	Java™ Enterprise Edition
API	Application Programming interface (Interfaz de programación de aplicaciones)
EJB	Enterprise Java Beans
POJO	Plain Old Java Object
ORM	Mapeo objeto - relacional
XML	eXtensible Markup Language (lenguaje de marcas extensible)
SOAP	Simple Object Access Protocol (Protocolo Estándar de comunicación mediante datos XML)
Widget	Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños.
MVC (Modelo Vista Controlador)	Patrón de arquitectura de software que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y las comunicaciones.
SPA	Single Page Application. Aplicación cuyo tiempo de respuesta es más rápido que el de una aplicación web tradicional.
URI	Del inglés Uniform Resource Identifier es una cadena de caracteres que identifica los recursos de una red de forma unívoca.
Hipermedios	Conjunto de métodos o procedimientos para escribir, diseñar o componer contenidos que integren soportes tales como: texto, imagen, video, audio, mapas y otros soportes de información emergentes, de tal modo que el resultado obtenido, además, tenga la posibilidad de interactuar con los usuarios.
Widget	Pequeña aplicación o programa, usualmente presentado en archivos o ficheros pequeños.

